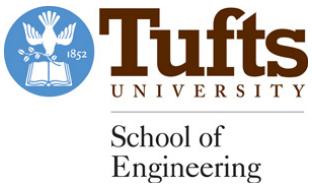
Machine Learning in Health Informatics: Making Better use of Domain Experts



Byron C. Wallace
Department of Computer Science
Tufts University

A dissertation submitted for the degree of
Doctor of Philosophy (PhD)
August 2012

Abstract

We present novel machine learning and data mining methods that make real-world learning systems more efficient. We focus on the domain of clinical informatics, an archetypical example of a field overwhelmed with information. Due to properties inherent to clinical informatics tasks – and indeed, to many tasks that require specialized domain knowledge – 'off-the-shelf' machine learning technologies generally perform poorly in this domain.

If machine learning is to be successful in clinical science, novel methods must be developed to: mitigate the effects of class imbalance during model induction; exploit the wealth of domain knowledge highly skilled domain experts bring to the task; and to induce better models with less effort (fewer labels). We present new machine learning methods that address each of these issues, and demonstrate their efficacy in the task of abstract screening. In particular, we develop new theoretical perspectives on *class imbalance*, novel methods for exploiting *dual supervision* (i.e., labels on both instances and features), and new *active learning* techniques that address issues inherent to real-world applications (e.g., exploiting multiple experts in tandem). Each of these contributions aims to squeeze better classification performance out of fewer labels, thereby making better use of domain experts' time and expertise.

The immediate aim in this work is to reduce the workload involved in conducting *systematic reviews*, and to this end we demonstrate that the developed methods can reduce reviewer workload by more than half, without sacrificing the comprehensiveness of reviews (i.e., without missing any relevant published evidence). But this is only an exemplary task; the approaches presented here have wider application to many real-world learning problems, i.e., those that require specialized expertise, exhibit class imbalance (and asymmetric costs) and for which limited human resources are available. We show that the methods we have developed bring substantial improvements over

previously existing machine learning approaches in terms of inducing better models with less effort.

To Deenie Wallace, who let me play video games when I was growing up.

And to Gary Wallace, who still knows more about computers than I do.

Acknowledgements

It is a trite gesture to offer acknowledgements for at the outset of one's thesis, but I'm going to do it anyway.

I am grateful first of all to my advisor, Carla Brodley, who encouraged me to pursue a Ph.D in spite of my initial reticence. She has been a fantastic advisor, offering just the right level of supervision to guide but not constrain. Carla has imparted to me her enthusiasm for tackling inter-disciplinary problems with real-world import. In addition to her cleverness and creativity, she also happens to be an awesome person.

I am also thankful that Carla recruited Kevin Small as a post-doc. Kevin is a bi-pedal Wikipedia of machine learning; the breadth of his knowledge is staggering at times. He has deeply influenced my views on research. Having conceived and taught a course alongside him, I can also say with confidence that he is a dedicated and thorough instructor. He is also someone with whom I can just as easily talk politics as matrices.

Of course, I am greatly indebted to my Committee members Roni Khardon, Anselm Blumer and Jaime Carbonell. Their thoughtful feedback has greatly improved this thesis. From Roni I learned the fundamentals of machine learning. Anselm provided careful technical proofs and corrections of this work, for which I am grateful. I am honored to have had Jaime on my committee; he is a pioneering machine learning researcher whose work I have always greatly respected.

This work is a result of a serendipitous collaboration with the Tufts Evidence-based Practice Center, whose members I am lucky to have had as colleagues and mentors. Joseph Lau is impressively forward-thinking: his ideas are consistently 20 years or so ahead of their time. I hope this work is a small step toward bringing some of his long-standing ideas on artificial intelligence and medical informatics to fruition. Christopher Schmid, meanwhile, is a formidable statistician and a brilliant teacher. He is also a genuinely nice guy.[1] Issa Dahabreh is a joy to work with: his combination of cleverness and indefatigability is without match (and humbling).

Lastly, I am impossibly indebted to Thomas Trikalinos. Tom has served as a close mentor to me over the years, and has always made more time for me than he has. Tom's ability to instantly comprehend problems far removed from his home disciplines is amazing. All hyperbole aside, he is brilliant, and an unassailable scientist. And he brings an infectious enthusiasm to everything he works on. He has also become a close friend. I am extremely fortunate to have the

[1] And he makes an excellent pesto, to boot.

opportunity to continue working with him at Brown in the coming years.

On a more personal note, I would like to thank my long-standing roommate Subie Patel, who has always been willing to entertain technical discussions and even occasionally read paper drafts. And, of course, Meg: this whole endeavor would have been much more stressful without relaxing evenings with you. Thank you.

sometimes I laugh at victory
kissing these little question marks.

Aesop Rock

Contents

List of Figures

List of Tables

1

Motivation and Preliminaries

In this section we first introduce our motivating task of citation screening for systematic reviews, which unbiasedly appraise all of the evidence pertaining to a specific clinical question. These reviews play a critical role in informing medical practice, but are extremely laborious to conduct; we look to reduce this workload via data mining. By presenting the motivating problem at some length we aim to provide a context for the data mining contributions made in this thesis. However, while the obstacles and opportunities that have arisen in the citation screening application have motivated our methodological developments (30), we emphasize that the problems we discuss, and the solutions we propose, are widely applicable. Indeed, the characteristics that make citation screening difficult from a data mining perspective – class imbalance, asymmetric costs, pricey experts with limited resources, multiple annotators of varying cost and expertise – are inherent to many real-world problems, particularly in the clinical domain.

After introducing systematic reviews in Section 1.1, we review machine learning fundamentals in Section 1.2 for the uninitiated. In Section 1.3 we outline open machine learning problems inherent to the citation screening task. These issue are common to real-world learning scenarios, and addressing them is thus imperative if machine learning methods are to be of practical use. The remainder of this thesis will be concerned with doing just that.

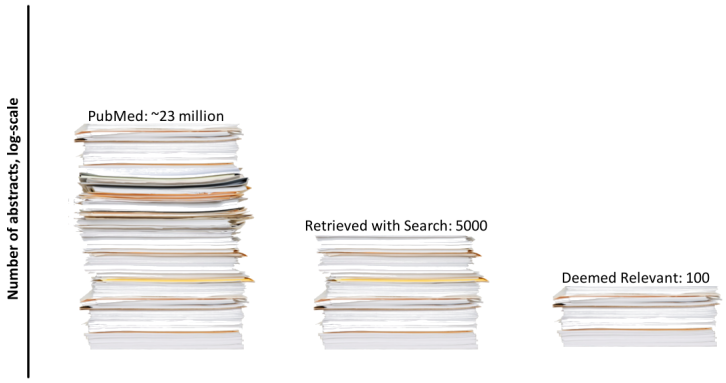


Figure 1.1: The citation screening process.

1.1 Systematic Reviews

Systematic reviews are increasingly used to inform all levels of healthcare, from bedside individualized decisions to policy-making. A systematic review addresses a precisely formulated clinical question by following a protocol of well-defined steps (17, 46). To minimize selection bias, systematic reviews appraise and analyze all research reports that fulfill a set of pre-defined eligibility criteria. To identify all eligible reports, reviewers conduct broad searches of the literature and then manually *screen* the retrieved citations for eligibility, i.e., read each abstract to decide if it meets the inclusion criteria.

All relevant (potentially eligible) citations are then reviewed in full-text to select those to be ultimately included in the systematic review. We refer to this latter step as *level-2 screening*, and the initial abstract screening step as *level-1 screening*. We will return to this distinction when evaluating methods for semi-automating screening. The citations ultimately deemed eligible for inclusion – those that pass level-2 screening – are then summarized in the review. Ideally, this is done quantitatively via meta-analysis (52), i.e., statistical pooling of the results reported in the individual studies. Performed correctly, meta-analysis can provide better a estimate of treatment efficacies than any individual clinical study, due to its increased statistical power (97). Meta-analyses are considered the strongest form of evidence, and are therefore a cornerstone of modern evidence-based medicine. Our focus is on mitigating the workload required to winnow the overwhelming amount of published clinical literature down to the tens of studies to be distilled into usable medical knowledge.

Figure 1.2: The abstracts manually screened by reviewers for a systematic review conducted at the Tufts Evidence-based Practice Center.

Screening citations for systematic reviews is a tedious, time-consuming and critical step in the evidence synthesis process. Failure to identify eligible research reports threatens the validity of a review. Typically, reviewers screen around 5,000 citations for eligibility, approximately 100 of which are deemed relevant and subsequently reviewed in full text. Of these, at most a few dozen are ultimately included in the systematic review. This is depicted in Figure 1.1, which plots on a log-scale the number of citations at each step in the described winnowing process (the right-most stack represents those citations screened in at the level-1 level). Much larger projects are not uncommon. For example, in a project that involved three evidence reports conducted for the United States Social Security Administration on the association of low birth weight, failure to thrive, and short stature in children with disability, the Tufts Evidence-based Practice Center screened over 33,000 abstracts (44, 125, 171). Figure 1.2 illustrates the amount of labor involved in the screening process: all of the abstracts that comprise the stacks in this photo were read by an expert.

An experienced reviewer can screen an average of two abstracts per minute. At this rate, a project comprising 5,000 abstracts requires 5 person days (40 hours) of uninterrupted work time. Moreover, abstracts for difficult topics may take several minutes each to evaluate, increasing by several fold the screening time. In total, a systematic review with meta-analysis can take between 1,000 and 2,000 person hours. Part of this time appears to be related to topic refine-

ment and setup; the rest depends on the number of included papers (2). The experts conducting such reviews are typically medical doctors, whose time is obviously expensive. Nationally, then, the cost of producing systematic reviews is tremendous.

These costs are only going to increase. Systematic reviews have gained wide acceptance as a practical way to provide reliable and comprehensive syntheses of the expanding medical evidence base. MEDLINE indexes more than 20,000 new randomized trials from 2010 alone, and the increasing trajectory of publication rates shows no signs of slowing. It is increasingly difficult to keep up with new information for both performing new reviews and updating existing reviews (18). Exacerbating the challenge of information overload, the standards for systematic reviews and meta-analyses are more demanding now than they were only ten years ago. The time to complete a systematic review and meta-analysis has not decreased over the last three decades. Indeed, the US Agency for Healthcare Research and Quality's comparative effectiveness reviews take at least 13 months to complete, an amount of time that has grown consistently during the last 15 years.

Yet researchers must continue producing such reviews, as they are critical to informing medical best-practice. Machine learning methods are plainly needed to reduce the workload involved in conducting systematic reviews. But as we will discuss, off-the-shelf techniques are not up to the task, due in part to unrealistic assumptions often made in machine learning. The citation screening task can thus be viewed as an exemplar problem that brings to the fore open machine learning questions. Solutions to these problems beget an immediate reward – namely reducing workload in systematic reviews – but are also of import to real-world data mining, in general.

In this thesis we consider the citation screening task from a machine learning (ML) vantage, re-casting it as a *classification* task. Specifically, we look to induce a model capable of discriminating 'relevant' from 'irrelevant' citations for a given review. The idea is to acquire minimal supervision from the participating experts (i.e., have them label a small subset of the citations retrieved via their broad literature search), induce the classification model, and then apply it to

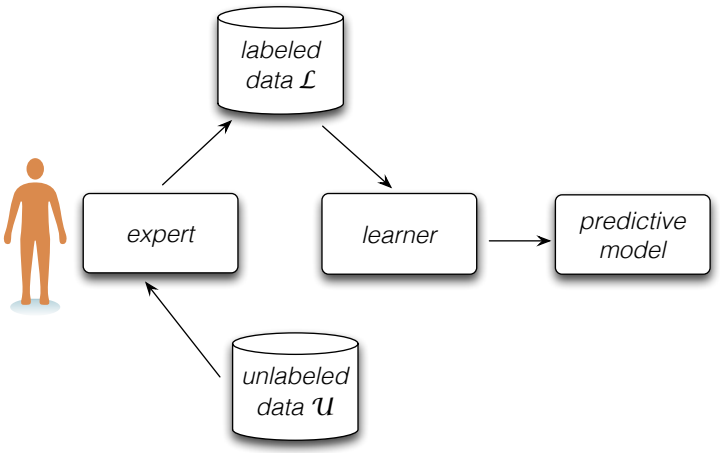


Figure 1.3: The supervised learning paradigm. The (human) expert labels a sample of the data to be classified, and this labeled data is used to induce a classification model.

the remaining citations. The reviewers will trust the model's exclusion decisions, and thus will not need to screen those citations the classifier deems 'irrelevant', thereby mitigating workload.

For completeness, we next introduce machine learning – we advise readers familiar with basic ML concepts (in particular: supervised learning, text classification, and Support Vector Machines) to skip this section. In Section 1.3, we identify open problems in machine learning that render 'off-the-shelf' methods insufficient for the task of citation screening, motivating the remainder of this thesis.

1.2 Supervised Machine Learning

The problem of classification falls under the umbrella of *supervised machine learning* methods. Generally speaking, these methods *train* a *learner* (model) on a set of labeled data with the aim of subsequently predicting the (unknown) labels of novel instances. This training step involves estimating, or 'learning', the parameters of the selected model. The supervised learning paradigm is depicted schematically in Figure 1.3.

As an example, consider a common benchmark task in machine learning: predicting the price of houses given their characteristics (75). It is reasonable to assume that housing prices are a function of their attributes; e.g., number

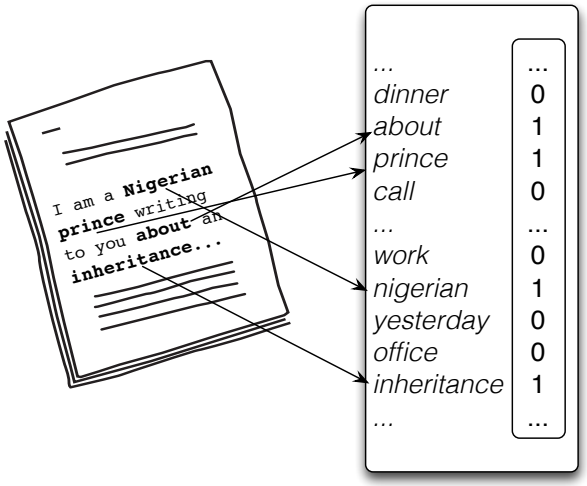


Figure 1.4: The (binary) Bag-of-Words (BoW) representation.

of bedrooms, square footage, and so on. In the parlance of machine learning, such attributes are referred to as *features*. Each *instance* (house, in this case) is then represented by a *feature-vector* that encodes these attributes. In the case of a simple linear model, we then might assume that features relate to cost as specified by Equation 1.1.

$$cost(house) = \beta_0 + \beta_1(\text{number of bedrooms}) + \beta_2(\text{square footage}) + ... \quad (1.1)$$

In this case, 'learning' involves estimating the coefficients (i.e., the βs) from a training set comprising some number of houses, including their attributes and their costs. To estimate our parameters, we would simply regress the former against the latter. This process is considered *supervised* because the model is induced over *labeled* data. Here, the label for a given house is its cost. This is an example of a *regression* task, because the target variable in this case – cost – is continuous. This is in contrast to classification tasks, wherein we aim to predict discrete *category* (class) to which an instance belongs.

1.2.1 Text Classification

Consider a task more closely related to that of citation screening: spam classification. In this case, instances are e-mails, and the aim is predict whether

a given e-mail is spam (or not). Before training a model, the e-mails must be transformed into a suitable representation, i.e., mapped into a feature-space. The most commonly used representation for text classification tasks is called Bag-of-Words (BoW) (98). This is an unstructured representation in which every word that appears in a corpus is assigned a unique index. Subsequently, each document is represented by a vector comprising **1**'s at the indices corresponding to words that appear in it and **0**'s elsewhere.

This is referred to as a binary BoW representation, and we have depicted the representation in Figure 1.4. The entries in the vector corresponding to the observed (bolded) words are set to **1**; the unobserved words (e.g., 'dinner') were present in other e-mails, but not this one. Finally, uninformative words, such as 'I' and 'a', are simply ignored. These are sometimes referred to as *stop words* (85).

Binary BoW is the simplest of the BoW representations. More sophisticated variants exist, e.g., term-frequency/inverse-document-frequency (TF/IDF) (110), which weights terms as a function of their frequency in the corresponding document relative to its overall frequency throughout the corpus. In our experience, however, the specific BoW variant has little effect on performance; in this work, all BoW representations are binary.

Once documents are mapped into a feature-space representation, one can apply machine learning algorithms to induce classifiers that will predict to which category (e.g., spam/not-spam) examples belong, given their feature-vector. Many learning algorithms exist, but in this work we will rely primarily on Support Vector Machines (SVMs) (148), which we review in the following section. SVMs are well-suited to the citation screening task because they are particularly adept at text classification (84). That said, many of the machine learning contributions in this thesis are independent of the underlying learning algorithm.

1.2.2 Support Vector Machines (SVMs)

The Support Vector Machine (SVM) is a state-of-the-science classifier (31). Intuitively, it works as follows. Given a training dataset comprising instances (feature vectors) and their labels (presumably provided by a domain expert), we look to

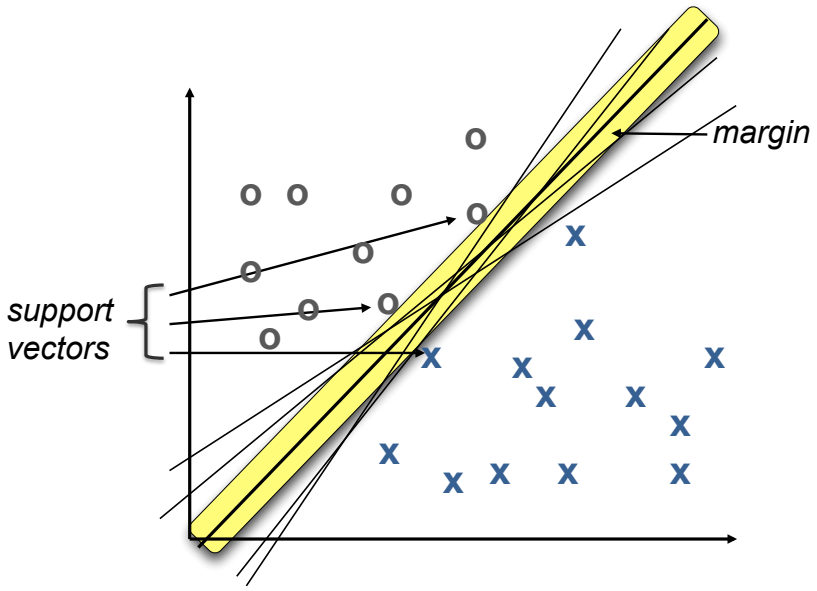


Figure 1.5: The max-margin approach favored by Support Vector Machines.

find a hyper-plane[1] that separates the instances into their respective classes as accurately as possible. New instances are classified according to the side of this hyper-plane on which they fall. Non-linear decision surfaces, i.e., cases in which the instances comprising the respective classes cannot be separated by a simple line, can be accommodated by implicitly mapping feature vectors to a higher dimensional space in which linear separation is feasible. This is referred to as the *kernel trick*.[2]

For any given dataset, there is usually an infinite number of hyper-planes that separate the data equally well. SVMs take a *max-margin* approach to picking one of these: we select the hyper-plane that maximizes the distance between the nearest instances from the respective classes (the *support vectors*). This intuition is perhaps best grasped via an illustration: consider Figure 1.5. In this simple two-dimensional problem, there are two classes, the **o**'s and the **x**'s. The max-margin approach is intuitively agreeable – you want members of both classes to be as far from the decision boundary as possible. It is also theoretically motivated by statistical learning theory (159). Of course, data will not always be separable. In such cases, there will be a trade-off between maximizing the margin between classes and correctly classifying the training data.

[1] A hyper-plane is just a generalization of a line to many dimensions.
[2] A thorough treatment of kernel methods is beyond the scope of this work; cf. (140).

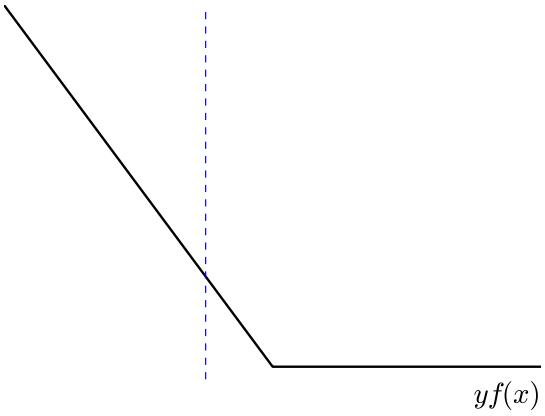


Figure 1.6: Hinge-loss. The y-axis is loss; the x-axis is $yf(x)$, where $f(x)$ generally encodes some confidence measure in the prediction. When $yf(x) > 0$, a correct prediction has been made. We have demarcated a hypothetical threshold for this by the dotted line. Note that immediately to the right of this line loss is still incurred, despite a correction prediction being made. This effectively penalizes low-confidence predictions, i.e., encourages a large-margin in classification.

More precisely, the preceding intuition can be operationalized by optimizing the following objective function:

$$\underset{\mathbf{w},b,\xi}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \tag{1.2}$$

$$s.t. \quad y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \xi_i \;\; \forall i = 1\ldots m \tag{1.3}$$

$$\xi_i \geq 0 \qquad\qquad \forall i = 1\ldots m \tag{1.4}$$

where we denote the vector characterizing the hyperplane by $\mathbf{w}$, instances by $\mathbf{x}_i$, and an intercept (or bias term) by b. The curious $\frac{1}{2}$ is just a mathematical convenience. Further, we define $\xi \in [0,\infty)^m$ as a slack variable vector to minimize instance-wise hinge-loss, which is defined by:

$$l(y, f(x)) = max(0, 1 - y \cdot f(x)) \tag{1.5}$$

Hinge-loss is depicted in Figure 1.6. This loss function is well-suited to SVMs because predictions that violate the margin incur loss, even if the classification is correct. We note, however, that hinge-loss is just one of many available loss-functions, all of which penalize empirical error in different ways.

Finally, C is a tradeoff parameter between misclassification error and regularization.[1] Equations 1.2 through 1.4 thus codify the max-margin and error-minimization principles. This constitutes a *quadratic program* and can be solved using standard optimization procedures (174).

The preceding machinery is sufficient to perform text classification: one first maps documents into BoW representation and then induces an SVM (or some other inductive classification model) on the training set. Having reviewed in brief the basic machine learning technologies on which we will rely, we now turn to the open research problems in ML addressed in this thesis.

1.3 Advancing Machine Learning Through its Application: Thesis Contributions

At first glance, it may seem tempting to conclude that existing machine learning technologies are sufficient to semi-automate citation screening, and that we should therefore simply use one of the many available ML software packages (e.g., Weka (73)) and be done. A few characteristics, however, make the citation screening task challenging (and thus interesting) from a machine learning vantage. We also re-iterate that these characteristics are common to real-world learning tasks, in general; thus while our contributions are motivated by the citation-screening task, they also have more general import.

A pervasive issue in our task is extreme class imbalance: there are far fewer relevant than irrelevant citations in any given systematic review. Moreover, the misclassification costs are asymmetric. It is imperative that researchers undertaking a review identify *all* potentially eligible studies, i.e., false negatives are costlier than false positives. Learning under imbalance with asymmetric misclassification costs is a common problem in deployed machine learning, and has thus received quite a lot of attention in the literature of which there are at least three surveys (71, 77, 83). Somewhat surprisingly, however, there is little theoretical understanding of the issue. Practitioners are thus provided with little guidance when faced with an imbalanced dataset. We present a theoretical per-

[1] Regularization is a strategy to avoid over-fitting, generally by attempting to keep the model simple, e.g., keeping most weights small.

spective on class imbalance in Chapter 2. This probabilistic framework motivates our approach of bootstrap-aggregating (*bagging*) classifiers induced on balanced samples of the training data (166). Using similar reasoning, we show that class probability estimates can also be improved via bagged/undersampled estimates (161).

There is also a fundamental issue of making better use of domain expert's time and expertise. In particular, the underlying target concept is different for every systematic review, and thus a new classification model must be induced for *each* new review topic. This imposes a substantial burden on domain experts, as they must label (screen) a sufficiently large training corpus for each review. Furthermore, the minimum level of biomedical expertise required for this labeling task precludes the outsourcing of annotation work to low-cost crowd-sourced services such as Mechanical Turk (`http://www.mturk.com`). As a means of reducing the burden on domain experts (and hence cost), we exploit the *dual supervision* and *active learning* frameworks. In the former, one exploits supervision on features in addition to instance-labels to expedite training. In the latter, the learning model interactively selects for labeling the (unlabeled) instances that are most likely to be informative, i.e., helpful in inducing a classification model.

First consider dual supervision. It is natural to ask whether domain experts can impart knowledge directly to the model, rather than indirectly via instance labels alone. The dual supervision framework allows just this. In dually supervised learning, experts annotate specific features (here, terms) that correlate with the respective classes of interest (relevance/irrelevance; spam/not-spam). This is in contrast to traditional supervised learning algorithms, which exploit only *instance labels*. These associate with each training example a single class label. However, domain experts may be able to provide more direct forms of supervision. For example, if one is inducing a model to discriminate positive from negative movie reviews, the presence of the word 'great' is likely to indicate membership in the former class, while the word 'terrible' suggests the latter. In Chapter 3 we propose a novel dually supervised approach that extends the Support Vector Machine (SVM) (45) model, which we call the Constrained

Weight-space SVM (CW-SVM) (151). In addition to binary labels on features, the CW-SVM can learn from ranked feature information. We show that this method outperforms baseline classification methods and previously proposed approaches to dual supervision.

Dual supervision attempts to make better use of experts by accommodating more direct supervision; active learning looks to achieve the same goal via different means. Specifically, in active learning the aim is to build a better model with fewer labels by picking the training instances cleverly, rather than at random. But as we shall demonstrate, off-the-shelf active learning methods (e.g., uncertainty sampling (158)) do not work well for our task. In Chapter 4 we develop novel active learning methods to expedite the training process. These methods are specifically designed for class-imbalanced scenarios because existing active learning methods perform poorly when applied to imbalanced datasets (165). We also address other issues in real-world active learning. For example, in the citation screening scenario a small number of experts (reviewers) typically participate in the screening task, some of whom are veteran reviewers, and others who are relatively new to systematic reviews. This is at odds with the usual assumption in (active) learning that there is a single, infallible oracle willing to provide labels at a fixed cost. To address these real-world problems we have developed active learning methods that perform instance allocation in multiple, imperfect labeler scenarios (167).

Another tacit assumption usually made in active learning is that labels for instances are of equal cost. In reality, of course, the cost of labeling instances will vary according to their difficulty, i.e., how long it takes an expert to categorize them. Exploiting this intuition, in Chapter 4 we develop a model to predict instance annotation time and incorporate this into the active learning process to select instances with high return on investment, i.e., that are likely to provide a lot of information at little cost (163).

In Chapter 5 we then combine the above two strategies, and present an algorithm for dually supervised active learning (165). We show that this strategy outperforms existing active learning methods, particularly in the case of imbalanced data. In Chapter 6 we present results from realistic experiments on

citation screening datasets using our methods. We show that these methods can indeed substantially reduce workload, without sacrificing the thoroughness of reviews. We also present *abstrackr*, our open-source web-based tool that implements these technologies, thereby making the machine learning tools available to systematic reviewers. We conclude in Chapter 7 by discussing our contributions and future research directions.

We have introduced the task of citation screening for systematic reviews, which will motivate the data mining problems tackled in the remainder of this thesis. The overarching aim is to squeeze better models from fewer labels. Beyond the immediate problem of citation screening, these issues are inherent to many real-world tasks in which machine learning has the potential to reduce human workload, and solving them thus has broad implications. In the remainder of this thesis, we address these problems in turn. We start by addressing the problem of learning under *class imbalance* in the following chapter.

2

New Perspectives on Class Imbalance

Class imbalance refers to the scenario in which the number of instances from each class is (perhaps extremely) unequal. For example, in the case of citation screening, there are far fewer relevant than irrelevant citations.[1] Imbalance is common in real-world learning tasks, e.g., detecting oil spills (93), text classification, and medical applications (41). The problem of imbalance is exacerbated by the fact that in imbalanced scenarios, the minority class is usually of primary interest. That is, misclassification costs are typically asymmetric so as to emphasize correct classification of minority instances. To consider two concrete examples: the majority of email is spam (154), but classifying legitimate email as spam is highly undesirable; most financial transactions are legitimate, but it is expensive to miss any instances of fraud (126).

Unfortunately, discriminative models induced over imbalanced datasets tend to fare poorly in terms of their predictive accuracy with respect to the minority class; such models generally suffer from low sensitivity (1).[2] Indeed, imbalance is problematic for machine learning methods in general, as it tends to bias inductive learning algorithms in favor of the majority class, resulting in false negatives. In our application this means wrongly designating relevant citations as irrelevant, a costly mistake because missing even one relevant citation may invalidate an

[1] It also naturally occurs in multi-class scenarios when one is interested in classifying instances as belonging to a specific class j versus not belonging to j.

[2] Sensitivity is also sometimes referred to as recall.

entire review (see Section 1.1). Mistakes in the other direction are less expensive: wrongly classifying an irrelevant citation as relevant incurs only the added cost of the time taken by an expert to subsequently exclude it from the review.

In this chapter we consider the problem of imbalance in the context of two machine learning tasks: classification (Section 2.2) and probability estimation (Section 2.3). The former is the standard classification task described in Section 1.2. The latter involves estimating the probability that a given instance belongs to a specific class, as opposed to simply predicting that it does or does not. Probability estimates are useful for providing a measure of confidence regarding a class prediction. For example, in our case we may wish to classify a citation as irrelevant only if we have a high confidence that it indeed is.

We will show that imbalance biases both classifiers and probability estimators, and provide theory as to why this is the case. Motivated by this exposition, we will propose solutions for both tasks that mitigate the effects of imbalance and produce less-biased classifiers and probability estimators for imbalanced data. Before presenting our work on the problem of imbalance, we next place our work in context by reviewing related work. We do not, however, attempt an exhaustive survey of the literature regarding imbalance; readers interested in a more detailed summary of existing methods for handling imbalance (especially for classification) should consult one of the existing surveys of the matter (71, 77, 83). We note that a portion of this chapter appeared in the *2011 Proceedings of the International Conference on Data Mining* (ICDM 2011) (166).

2.1 Related Work

The problem of imbalance in classification tasks has received considerable research attention (1, 36, 60, 77, 82, 83, 118, 170, 175). Techniques for mitigating the effects of imbalance fall into two categories: re-sampling methods (59, 93, 106, 156) and methods that alter the empirical error function being optimized over the training set to emphasize recall (105, 155, 175). Sampling-based methods re-sample the training set to make the class distribution more equal, either by undersampling majority instances or oversampling minority instances.

These two strategies are opposite means to the same end: making the training distribution roughly balanced.

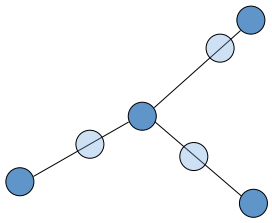


Figure 2.1: SMOTE (Synthetic Minority Oversampling TEchnique) (35), graphically. The darker instances represent observed minority examples. The lighter blue instances represent synthetically created minority examples. These are generated by interpolating the observed rare instances.

An interesting variant of oversampling is SMOTE (35), in which pseudo-minority instances are created by interpolating observed minority examples in feature-space. This is depicted in Figure 2.1. A variant of this strategy is *borderline*-SMOTE, in which only minority instances near the discriminating plane are used to generate pseudo-minority points (74). The hope is that placing additional minority points near the border will have a greater effect on the induced classifier.

Cost-based strategies increase the cost of false negatives relative to that of false positives during training, thereby favoring parameters that correctly classify minority instances. For example, cost-weighted SVMs decompose the empirical cost C (see Equation 1.2) into C_{FN}, C_{FP}, corresponding to costs for false negatives and for false positives, respectively – the former is usually set higher than the latter to reflect an emphasis on sensitivity (155). More generally, any learning algorithm that looks to minimize empirical error on a training set can take this approach by assigning different costs to mistakes made on instances from the respective classes in the objective function.

Empirically, it seems sampling-based strategies are more effective in mitigating the effects of imbalance – see (80) for an exhaustive empirical evaluation.[1] Undersampling, especially, often outperforms other methods. In the following section we will elucidate *why* this is the case, as the empirical success of undersampling was previously an open question. Subsequently aggregating an ensem-

[1]Obviously, what constitutes good performance depends on the metrics one is using. Loosely, we will assume that one is interested in inducing classifiers that perform at least as well on minority instances as they do majority instances, i.e., achieve sensitivity at least as high as specificity (see Equations 2.1 through 2.4).

ble of these classifiers induced over undersampled training sets is, in our view, a natural next step. Yet this *bagging* approach is often overlooked by researchers. Indeed, the most comprehensive empirical comparison of strategies to mitigate the effects of imbalance (80) did not include bagged classifiers induced on bootstrap (undersampled) training sets, despite undersampling performing the best of all methods, overall.

Elsewhere, researchers have investigated boosting-style (65) algorithms to mitigate the effects of imbalance. Recall that boosting is an iterative procedure in which the training set is re-sampled at each step to emphasize correct classification of those instances on which mistakes were made in the previous round. Once complete, boosting produces an ensemble of classifiers induced over the varying per-round distributions; final predictions are taken as an aggregate over these, where each classifier contributes to the overall prediction with weight proportional to its empirical performance. Methods that address imbalance via boosting typically work by re-sampling or otherwise modifying the training set at each round during boosting.

Seiffert et al. (141) report that standard boosting is, surprisingly, competitive with other techniques for handling imbalance. This is perhaps because it forces correct predictions on the minority instances in the training set. Chawla et al. (37), meanwhile, proposed SMOTEing the training dataset at each step in the boosting process. Liu et al. (106) proposed a similar strategy, in which they extend Schapire's classic AdaBoost algorithm (138) to induce a classifier on a balanced sampling of the training data at each round, producing a committee of classifiers each induced over independently drawn balanced sub-samples of the training data. This approach of bootstrap-aggregating, or bagging (26), classifiers trained on re-sampled subsets has been proposed independently in the literature several times (79, 87, 156), and we will return to it in Section 2.2. Guo and Viktor (70) propose mixing synthetic data and boosting. Specifically, at each round they identify hard instances from both the majority and minority classes. These are fed into SMOTE to generate pseudo-instances that are subsequently used to form a balanced dataset. The classifier trained on this training set thus equally emphasizes correctly classifying difficult instances from both classes.

In contrast to the case of classification, there has been little work investigating the reliability of class probability estimates in the context of imbalanced data, the task we address in Section 2.3. This is surprising because it is in such imbalanced cases that probability estimates could potentially be of most use: probability estimates can inform decision-theoretic models that look to make minimum-cost classifications in scenarios with asymmetric costs. More generally, probability estimates offer more granular information than class predictions alone. Indeed, due to their utility, there has been a substantial amount of work investigating attaining probability estimates from supervised learning *in general*; notably due to Niculescu and Caruana (120, 121) and Zadrozny and Elkan (179). We review the technical details of existing calibration methods in Section 2.3. The only work of which we are aware that investigates probability estimates specifically in the context of imbalance is due to Cieslak and Chawla, who investigated the specific case of Probability Estimation Trees (PETs) for imbalanced data (39). Their main focus was to elucidate the interaction between imbalance methods for PETs, and corresponding evaluation measures under circumstances where training and testing samples differ. By contrast, our work concerns calibrated probability estimates in general, as opposed to estimates produced directly by PETs. Moreover, we do not exclusively concern ourselves with scenarios in which the train and test sets differ.

Having reviewed in brief much of the imbalanced learning literature above, we next turn our attention to learning classifiers over imbalanced data. In particular we introduce a theoretical framework with which we analyze the problem, and using this we advocate the strategy of bagging classifiers induced over balanced bootstrap samples. In Section 2.3 we address the equally important, but under-studied, task of producing good class membership probability estimates for imbalanced data. After demonstrating that existing supervised learning methods for probability estimates perform poorly in imbalanced cases, we propose a method similar to the aforementioned bagging technique for classification.

2.2 Classification for Imbalanced Data, Redux

We first consider the problem of class imbalance from the perspective of classification. We approach the problem from a probabilistic perspective, and from this vantage identify dataset characteristics (such as dimensionality, sparsity, etc.) that exacerbate the problem. Motivated by this theory, we advocate the approach of bagging an ensemble of classifiers induced over balanced bootstrap training samples, arguing that this strategy will often succeed where others fail. Thus in addition to providing theoretical insight into the problem of class imbalance, corroborated by our experiments on both simulated and real datasets, we provide practical guidance for the data mining practitioner working with imbalanced data.

Classification under imbalance is an important problem in data mining due to the prevalence of imbalance in real-world tasks and the relatively poor performance achieved by existing learning algorithms on such datasets. Indeed, the problem of inducing classifiers over imbalanced datasets with asymmetric costs has been designated one of '10 challenging problems in data mining research' (177). The prevalence of the problem has motivated a significant amount of methodological research into learning under imbalance, much of which is reviewed above. There are several surveys on the topic of imbalance (71, 77, 83). Yet while many methods have been proposed to handle imbalance, there has been relatively little effort to elucidate the underlying mechanisms that cause discriminative models to fail when faced with imbalanced datasets. Because the conditions that lead to poor classifier performance under imbalance are not well understood, it is not clear which (if any) of the myriad existing algorithms for mitigating the effects of imbalance ought to be employed for a given task. Consequently, when faced with imbalance, the data mining practitioner is left with little guidance regarding how to proceed. Here we theoretically motivate and empirically justify the use of the simple undersampling strategy for imbalanced datasets under particular conditions (e.g., high-dimensionality). This work thus provides an explanation for the otherwise surprising observation that undersam-

pling tends often to outperform what are ostensibly more advanced techniques (e.g., SMOTE) (80).

While effective, undersampling is problematic because it is a high-variance strategy: classifiers induced over different bootstrap samples will sometimes have significantly different predictive performance. To ameliorate this problem, one can use the *bagging* (26) variance-reduction ensemble method. Bagging reduces classifier variance by creating an ensemble of predictors over independently drawn bootstrap training samples. The strategy of bagging classifiers induced over balanced bootstrap training sets has been independently proposed several times (e.g,. (79, 87, 106, 156)), but *why* and *when* it should outperform other methods has been largely unexplored. In this work we provide such an explanation, and we conclude that *in cases where data is imbalanced and either high dimensional, highly skewed or sparse,*[1] *practitioners should bag classifiers induced over balanced bootstrap samples.* Specifically, we contend that while algorithmic approaches to handling class imbalance often improve performance, sampling approaches such as undersampling will usually perform better. We show that cost-sensitive approaches that look to improve performance achieved under imbalance by, for example, modifying the relative costs of false negatives to false positives in an objective function, will still often induce biased classifiers.

The primary contributions of our work on classification under imbalance are as follows. We develop a probabilistic theory to quantify the effects of imbalance on the induction of empirical-loss minimizing models (e.g., SVMs). We show that under a few weak assumptions, such models will necessarily be biased toward the minority class, explaining the observed degradation in recall over test datasets. Furthermore, we decompose this bias into sub-components, some of which reflect properties of the training sample, and others that modify the empirical loss calculation. In light of this decomposition, we analyze several popular methods for handling imbalance, and discuss under what conditions one can expect them to work. We theoretically motivate, and experimentally demonstrate the efficacy of, the simple but robust strategy of bagging classifiers induced over

[1] By highly skewed we mean severely imbalanced. By sparse we refer to datasets comprising instances with a high proportion of 0 valued features.

balanced, bootstrap samples under various learning conditions. By providing a probabilistically motivated theory of imbalance, the implications of which are borne out both in our simulation and empirical experiments, we shed new light on a long-standing problem and provide disciplined guidance to practitioners facing imbalance.

2.2.1 An Analysis of Imbalance

In supervised classification, we are given an observed training set $\mathcal{D}$ over which a predictive model c is to be induced. Typically, c is constructed to optimize a specified objective function (equivalently, minimize some loss function) over the points comprising $\mathcal{D}$. More precisely, let us assume that given $\mathcal{D}$, the aim is to induce a linear classifier that minimizes the empirical error over $\mathcal{D}$.

Typically, however, one does not look to simply minimize the overall empirical error (i.e., maximize accuracy) in imbalanced scenarios. Consider, for example, that the trivial majority classifier that classifies every instance as belonging to the majority class will achieve 99% accuracy if the prevalence of the minority class is 1%. Rather, it is generally accepted that considering sensitivity and specificity separately – or taking a weighted combination of them, e.g. via the geometric mean of sensitivity and specificity (which is traditionally referred to as the g-mean) or F-score[1] – is more appropriate for imbalanced datasets (36, 83, 91, 128, 155, 170). These metrics are defined in Equations 2.1 through 2.5 where TP, FP, TN, and FN represent the number of true positives, false positives, true negatives and false negatives, respectively. In all of these metrics save for precision, prevalence drops out completely. Thus when one looks to maximize one of these, one is attempting to induce a classifier that performs well on instances from both classes, irrespective of prevalence. This insight will guide our analysis here. Note also that even when sensitivity and precision

[1]The F-score is often defined as a combination of precision and sensitivity, rather than specificity and sensitivity; we generally prefer using specificity in place of precision because it is independent of sensitivity, whereas precision is not. When we use specificity in place of precision, we shall indicate this with the superscript 'spec', i.e., using F_2^{spec}. In any case both formulations emphasize sensitivity regardless of prevalence.

(rather than specificity) are used, emphasis is placed on correctly classifying minority examples independent of the observed prevalence.[1]

$$\begin{aligned} \textit{sensitivity (recall)} &= TP/(TP+FN) && (2.1)\\ \textit{specificity} &= TN/(TN+FP) && (2.2)\\ \textit{precision} &= TP/(FP+TP) && (2.3)\\ F_2^{\text{spec}} &= \frac{5 \cdot \textit{sensitivity} \cdot \textit{specificity}}{4 \cdot \textit{sensitivity} + \textit{specificity}} && (2.4)\\ G\text{-mean} &= \sqrt{\textit{sensitivity} \cdot \textit{specificity}} && (2.5) \end{aligned}$$

The key assumption we will make in this work is that the observed positive and negative instances ($\mathcal{D}^+$ and $\mathcal{D}^-$) are drawn from two independent, latent distributions: P and G, respectively. Without loss of generality, we assume that positive instances constitute the minority class. Under this 'two-sample' assumption, it is readily apparent why a discriminative model induced over the sample $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$ achieves poor recall: the positive distribution P is under-represented and hence poorly characterized, while we are likely to have encountered 'outlying' negative examples due the comparatively large number of observations drawn from G. We are therefore likely to induce a separator that is skewed toward the minority class (i.e., closer to the minority points than it should be), resulting in poor predictive performance over hold-out instances from this class.

This intuition is illustrated by Figure 2.2, a synthetic example in which the $\times$'s represent the minority class and the ■'s the majority: the corresponding latent Gaussians (P and G) from which these samples were drawn are also shown. Here these distributions are unimodal; in such cases bias will be especially pronounced. We will not generally be making unimodal assumptions, however if P and G are dense around w^*, bias will be less of a problem because we will be increasingly likely to have observed points near this plane from both classes. The dotted line ($\hat{w}$) is the hypothesis induced over the training instances, while the

[1] Sensitivity and specificity are independent of prevalence in the sense that they are class-conditional proportions.

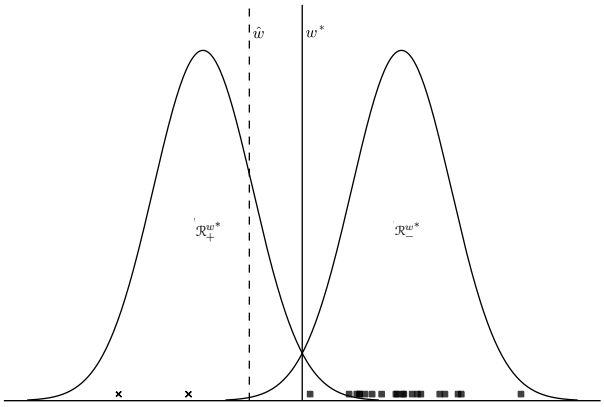


Figure 2.2: The bias of a linear separator induced over an imbalanced empirical sample in a one-dimensional example. Here the underlying distributions are shown, as well as a training sample comprising a few instances from the minority class (the $\times$'s) and ten times as many from the majority class (the $\blacksquare$'s). The solid line, w^*, is the optimal separator, w.r.t. the underlying distributions; i.e., this classifier will jointly maximize sensitivity and specificity over any draw from P and G. The dotted line, $\hat{w}$, is the max-margin loss-minimizing separator induced over the empirical sample. Note that $\hat{w}$ is biased toward the minority class, w.r.t. w^*. We also note that for certain measures of performance, $\hat{w}$ is indeed a better classifier than w^*. Indeed if the test sample is also imbalanced, $\hat{w}$ will result in a higher overall accuracy and is nearer to the Bayes optimal classifier than w^*. But again, we are not particularly interested in overall accuracy. This further illustrates the importance of metrics in classification for imbalanced datasets.

solid line (w^*) depicts the optimal separator of the underlying distributions, assuming we are interested in maximizing sensitivity and specificity over arbitrary draws from the latent distributions. In the example shown, the former is clearly skewed toward the minority class. We now formalize the intuition captured by Figure 2.2 more precisely.

2.2.1.1 The Bias of Empirically Estimated Separators

We begin with the 'two-sample' scenario introduced above. We restrict ourselves to the task of inducing a loss-minimizing separating hyperplane w that splits the input feature space into two half-spaces, $\mathcal{R}^w_+$ and $\mathcal{R}^w_-$. Instances that fall in the $\mathcal{R}^w_+$ region (left side of Figure 2.2) are predicted to belong to the minority (positive) class, and those in $\mathcal{R}^w_-$ to the majority (negative). Aside from empirical loss minimization over a training set $\mathcal{D}$, we make no further restrictions regarding the parameter estimation procedure. Indeed, there may be an infinite number of equivalent planes (i.e., loss-minimizing weight vectors) for a given training dataset;[1] we assume only that the selected $\hat{w}$ is one of these. As a notational convenience, we superscript $\mathcal{R}$'s with the planes that delineate them. We assume that the costs of false positives and false negatives are known, and denote these by $\mathcal{C}_{\text{fp}}$ and $\mathcal{C}_{\text{fn}}$, respectively.

Let us assume that the objective when training a classifier is to induce a separating plane w^* that maximizes per-class accuracy over arbitrary draws from the latent distributions P and G. We emphasize that this assumption is consistent with the standard metrics for classifier evaluation under imbalance, which are typically averages of rates. The most widely used of these metrics is perhaps the F-measure, which is a weighted harmonic mean of specificity and recall (Equation 2.4).[2] Another popular metric for imbalanced datasets is the geometric mean or G-mean (93), which is the square root of the product of class-wise accuracies, that is, sensitivity and specificity (Equation 2.5). By definition, these metrics are independent of the prevalence of the minority class, and thus one is

[1] These may be equivalent in terms of loss-minimization, however if we are using SVM, the 'optimal' (max-margin) solution will be unique.

[2] Note that when precision is used instead of specificity, F_2 is not independent of prevalence, because precision will depend on prevalence. However this metric still incorporates recall, which measures sensitivity to minority instances regardless of their prevalence.

tacitly ignoring the prevalence observed in $\mathcal{D}$. Maximizing these metrics therefore agrees with minimizing Equation 2.6. Alternatively, this objective can be viewed as learning under the minimax assumption, in which case we attempt to minimize the maximum loss under an arbitrary "covariate shift" (96).

In light of the preceding discussion, we define the optimal plane as follows

$$w^* = \underset{w}{\operatorname{argmin}}\, \mathcal{L}^{.5}(w) \tag{2.6}$$

where $\mathcal{L}^{.5}(w)$, the loss with respect to a balanced mixture of the latent distributions and is defined as follows:

$$\mathcal{L}^{.5}(w) = \mathcal{C}_{\text{fn}} \int_{\mathcal{R}_-^w} P(x)\mathrm{d}x + \mathcal{C}_{\text{fp}} \int_{\mathcal{R}_+^w} G(x)\mathrm{d}x \tag{2.7}$$

Ignoring the respective costs for a moment, the first term corresponds to $1-$*sensitivity*, and the second to $1-$*specificity*. In terms of minimizing these (i.e., maximizing sensitivity and specificity), w^* is thus the *ideal* separator with respect to the underlying distributions. The costs effectively allow one to weight sensitivity against specificity. Now consider the effect of imbalance. We denote the prevalence of the minority class in the training set by π (note that $\pi < .5$) and $\mathcal{D}_\pi$ to denote the distribution over all datasets drawn from P and G with minority prevalence π. Then the expected empirical loss of an arbitrary w is:

$$\mathbb{E}_{\mathcal{D}_\pi}[\mathcal{L}(w)] = \pi \mathcal{C}_{\text{fn}} \int_{\mathcal{R}_-^w} P(x)\mathrm{d}x + (1-\pi)\mathcal{C}_{\text{fp}} \int_{\mathcal{R}_+^w} G(x)\mathrm{d}x \tag{2.8}$$

We can also consider the empirical loss incurred over a particular dataset, $\mathcal{D}$:

$$\mathcal{L}_{\mathcal{D}}(w) \quad = \quad \frac{1}{|D|}(\mathcal{C}_{\text{fn}}|\{x|x \in \mathcal{D}^+ \wedge x \in \mathcal{R}_-^w\}| + \mathcal{C}_{\text{fp}}|\{x|x \in \mathcal{D}^- \wedge x \in \mathcal{R}_+^w\}|)$$

We denote by $\hat{w}$ a plane that minimizes empirical error over a particular draw from $\mathcal{D}_\pi$. We claim that minimizing the empirical loss will probably result in a plane that is skewed toward the minority class, with respect to w^*. We analyze the specific conditions under which this is the case, and show that the problem is exacerbated by imbalance. More specifically, we are interested in the conditions

under which the induced region delineating the positive instance space $\mathcal{R}_+^{\hat{w}}$ is smaller than the corresponding region induced by w^* with respect to the loss over the underlying distributions. When using inequalities (as in Equation 2.12) we will slightly abuse notation by implying the following scalars for $\mathcal{R}_+^w$ and $\mathcal{R}_-^w$:

$$\mathcal{R}_+^w = \int_{\mathcal{R}_+^w} P(x)\mathrm{d}x \tag{2.9}$$

$$\mathcal{R}_-^w = \int_{\mathcal{R}_-^w} G(x)\mathrm{d}x \tag{2.10}$$

$$\tag{2.11}$$

Then we can formalize our notion of bias as follows:

$$\mathcal{R}_+^{\hat{w}} < \mathcal{R}_+^{w^*} \tag{2.12}$$

Because we are inducing a classifier over an imbalanced training sample, the expected empirical loss-minimizing hypothesis will be biased toward P, with respect to w^*. This is because w^* minimizes $\mathcal{L}^{.5}$, i.e., loss with respect to draws from a balanced mixture of P and G, whereas $\hat{w}$ minimizes the empirical loss incurred on imbalanced samples.

Figure 2.3 provides intuition as to why this is the case via a simple one-dimensional example. The top plot shows the latent distributions (as before, we are arbitrarily assuming that P is the distribution to the left, i.e., the solid black line). Immediately below, we see the costs contributed by the two distributions for separators running along the x-axis. This is shown both respect to balanced samples, in which case the distributions contribute equally to the incurred cost, and with respect to imbalanced samplings. In the latter case the contribution of P (the minorities) is reduced by a constant.

We demarcate the point at which expected cost is at a minimum for the empirical loss, i.e., with respect to imbalanced samples, by the dotted vertical line; this corresponds to the expected $\hat{w}$ over an imbalanced training dataset. Similarly, we demarcate the loss minimizing point for the balanced case by the solid vertical line; this corresponds to w^*. We show these points on all three

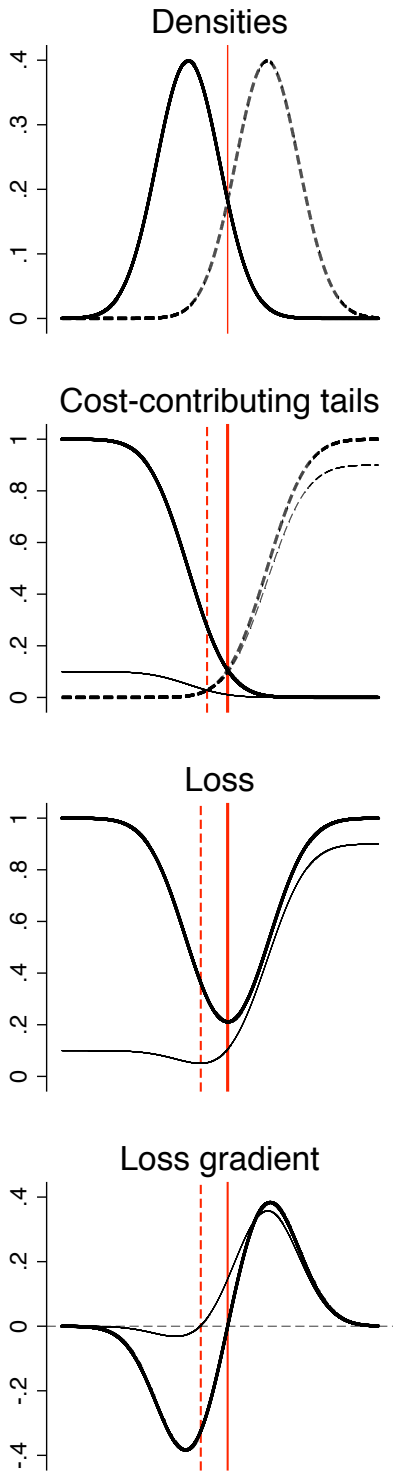


Figure 2.3: A graphical illustration of why linear separators induced on imbalanced datasets are biased, w.r.t. w^*; see text for discussion.

bottom plots. The corresponding losses are shown in the third plot (second from the bottom). The thinner line corresponds to the total expected loss in the empirical case; the thicker line to the loss incurred w.r.t. both distributions over balanced samples. Finally, we show the gradient of the loss in the bottom plot: this is at 0 when loss is at a minimum. We see that this loss is at minimum (the gradient of the loss is at 0) at a biased point, i.e., a $\hat{w}$ satisfying Equation 2.12.

We can quantify when we should expect the induced classifier ($\hat{w}$) to be biased. In particular, for latent distributions P and G, and training datasets with minority prevalence π, the expected loss given by Equation 2.8 suggests that we will probably induce a biased separator when:

$$(1-\pi)\mathcal{C}_{\text{fp}} \int_{\mathcal{R}_{+}^{w^*}} G(x)\mathrm{d}x > \pi\mathcal{C}_{\text{fn}} \int_{\mathcal{R}_{-}^{w^*}} P(x)\mathrm{d}x \tag{2.13}$$

That is, when w^* would incur a greater empirical cost than some alternative hypothesis w^γ because of the disproportionate contribution of false positives to this cost. In such cases, shifting w^* toward the minority class will reduce the empirical cost over $\mathcal{D}$, giving rise to a biased, empirical loss-minimizing hypothesis. In the following few sections, we will discuss methods for handling imbalance in light of this view of bias.

2.2.2 Why Weighted Empirical Cost Minimization is not Sufficient

Equation 2.13 decomposes the likelihood of inducing a biased separator into three sub-components: prevalence (π), costs quantifying mistakes made on instances belonging to the respective classes, and (latent) distributional characteristics. It would seem that the straight-forward strategy to handling imbalance, then, would be to fiddle with the $\mathcal{C}_{\text{fp}}$ and $\mathcal{C}_{\text{fn}}$ variables – in particular to penalize false negatives more heavily than false positives, or otherwise modifying the objective function to achieve this implicitly. We will refer to the family of methods that attempt to mitigate the effects of imbalance by assigning different costs to false positives/negatives during induction as *weighted empirical cost minimizing*

learners. Many methods of this type have been proposed in the literature, e.g., (106, 175).

However, modifying the empirical cost structure will often have no effect at all. In particular, *if the instances comprising the classes in the training dataset are separable, modifying the cost of false negatives relative to that of false positives in the objective function will not reduce bias.* This is trivially true; increasing the cost of false negatives will not budge the induced $\hat{w}$ if there are none in the first place.[1]

One can quantify the conditions under which modifying the empirical cost of false negatives/positives will be effective. Consider that this can reduce bias (Equation 2.12) if and only if it affects the loss incurred over $\mathcal{D}$ (Equation 2.9). For the moment, let $\mathcal{C}_{\text{fp}} = \mathcal{C}_{\text{fp}} = 1$. Denote the empirical loss-minimizing plane induced in this case by $\hat{w}_1$. Increasing the cost of a false negative to β times that of a false positive will produce a different plane only if there exists a point closer to the majority half-space than $\hat{w}_1$, i.e., if $\hat{w}_1$ results in at least one false negative. If no such point exists, $\hat{w}_1$ will already be loss-minimizing, regardless of β.

Fixing $\hat{w}_1$, and assuming $\pi|\mathcal{D}|$ minority samples, the probability that such a point will have been observed in $\mathcal{D}$ is

$$1 - (\int_{\mathcal{R}^+_{\hat{w}_1}} P(x)\mathrm{d}x)^{\pi|\mathcal{D}|} \tag{2.14}$$

As the degree of imbalance increases (i.e., π decreases), the probability that using weighted cost minimization over imbalanced training samples to counter imbalance will be effective in reducing bias decreases. Equation 2.14 also suggests that as the size of the training set increases, such strategies will become more effective, in general. Both of these observations are borne out in our simulation experiments (Section 2.2.4). The characteristics of P will also contribute to the (in)effectiveness of cost-sensitive induction procedures, e.g., if P happens to be dense around the plane in X defined by w^*, then weighting will improve classifier performance. By contrast, in the case of unimodal distributions (as depicted in

[1] We note that in the case of SVMs, attempting to explicitly induce an asymmetric margin (making it large for minority instances) (176) may mitigate this problem.

Figure 2.2) draws from the tails of the distribution will likely be rare, and bias will be pronounced.

2.2.2.1 Remarks on SMOTE

One of the most popular strategies for countering imbalance is the Synthetic Minority Oversampling TEchnique (SMOTE) (35). SMOTE is ostensibly a sampling strategy, insofar as it ultimately produces a balanced dataset on which to induce a model, but we argue that with respect to imbalance, SMOTE behaves similarly to the weighted empirical cost minimizing learners discussed above. SMOTE works by interpolating the observed minority instances with one another to create 'new', synthetic minority instances. In particular, this is done as follows. For each minority instance x^i, find the k minority points in $\mathcal{D}$ to which it is nearest. Now create synthetic minority points from x^i by selecting one of these neighbors x^n at random and creating a value for each feature j that falls on a random point along the line connecting x^i_j and x^n_j.

Due to the interpolation mechanism for creating synthetic instances, no pseudo-minority point produced via SMOTE will ever be located outside of the convex hull enclosing the observed minority instances. This observation implies that the probability that SMOTE will reduce bias during induction over an imbalanced dataset is similar to that for weighted empirical loss minimizing techniques (Equation 2.14), i.e., SMOTE should work in cases that weighted empirical loss minimizing methods work.

2.2.3 The Case for Undersampling and Bagging

We will now present arguments in favor of the undersampling plus bagging strategy for mitigating imbalance in light of the preceding discussion.

2.2.3.1 Why Does Undersampling Work?

The idea of throwing away most of one's data in order to induce a model seems anathema to statistical inference, as generally the best strategy is to exploit all available information. In spite of this, undersampling has proven effective in the case of imbalance, more often than not outperforming more advanced

methods (59, 80, 93, 106). The notion of 'outperforming', of course, pre-supposes a metric of interest. Most of the empirical work in the literature on learning under imbalance uses a weighted harmonic mean of recall and specificity (or recall and precision), and we will follow this convention here. Indeed, the emphasis on these metrics has motivated our interest in the balanced loss defined by Equation 2.7. Generally it assumed that recall is more important than overall accuracy on the majority class; how much so will depend on the task at hand. In cases that recall is emphasized more than recall, we might modify Equation 2.7 accordingly, which would increase the bias of empirical loss-minimizing separators.

Undersampling is effective despite its simplicity because it reduces the probability that the induced separator will be biased. More specifically, consider the inequality expressed in Equation 2.13, which quantifies the condition under which we are likely to induce a biased $\hat{w}$. Removing majority instances from $\mathcal{D}$ until $|\mathcal{D}^+| = |\mathcal{D}^-|$ effectively removes π from this equation. Thus, for a separator induced over an undersampled dataset, the condition under which we expect a biased plane becomes:

$$\mathcal{C}_{\text{fp}} \int_{\mathcal{R}_+^{w^*}} G(x)\mathrm{d}x > \mathcal{C}_{\text{fn}} \int_{\mathcal{R}_-^{w^*}} P(x)\mathrm{d}x \tag{2.15}$$

Crucially, this removes the imbalance component from the inequality (it becomes π on both sides). To illustrate the effects of this, recall the toy example depicted in Figure 2.2, in which training instances are drawn disproportionately from two latent one-dimensional Gaussians. In Figure 2.4, we draw 10 planes induced over balanced samples taken from the training set. All of these are less biased (closer to w^*) than the separator induced over the entire training dataset ($\hat{w}$). However, one can see that this is also a high-variance procedure – different re-samplings induce very different planes. We will now discuss how to mitigate this property via bagging (26).

2.2.3.2 Bagging for Imbalance

Bagging is a method of aggregating classifiers induced over independently drawn *boostrap* samples (26). *Bootstrapping* is a sampling mechanism that has tradi-

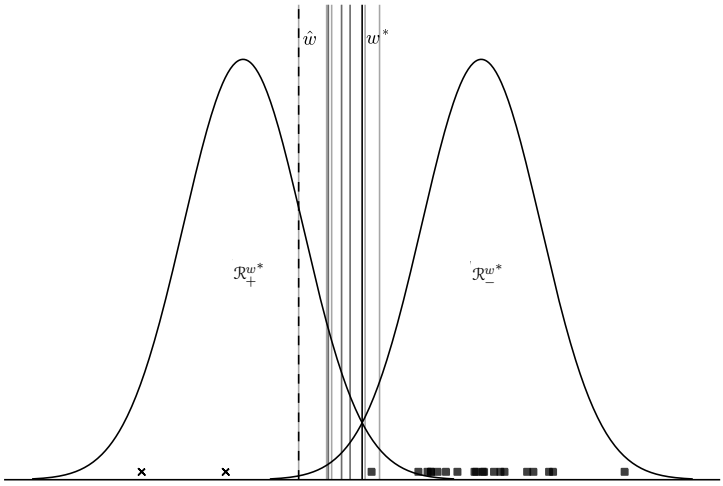


Figure 2.4: The effect of undersampling on separator induction. $\hat{w}$ is the (biased) plane induced over the entire dataset, w^* the optimal plane (w.r.t. the underlying distributions). The light grey lines depict the lines induced over independently drawn balanced bootstrap samples of the training data. Note that all of these are less biased (nearer w^*) than $\hat{w}$.

tionally been used to estimate the (true) standard error of a summary statistic calculated over an empirical sample $\mathcal{D}$ by calculating this statistic over n independently drawn 'bootstrap' samples taken from $\mathcal{D}$.

Bagging is a natural extension of the bootstrapping technique for predictive models that works as follows. We build an ensemble comprising B models, each induced over a bootstrapped sample of the training data. When a new instance is to be classified, each model makes a prediction, and the final, aggregate prediction is taken as the majority vote. Typically, bootstrap samples are drawn at random with replacement and i.i.d. from the original sample (62), and thus reflect the distributional characteristics of the original dataset. In our case, this would mean each sample would be imbalanced. This is undesirable because it would create bootstraps equally likely to induce biased classifiers. Indeed, the bagging methods proposed for imbalance advocate taking balanced samples, with the exception of Hido and Kashima (79), who propose 'roughly balanced' samples as a 'better motivated' (statistically) approach. However, given that we are aiming to approximate the boundary separating P and G, balanced bootstrap sampling is a more appropriate approach here.

In particular, consider that classifier induction is an instance of the boot-

strapping *two-sample* case, described by Efron (62). We observe a sample $\mathcal{D}$ drawn from P and G, disproportionately. We are interested in estimating a separator w.r.t. these distributions, independent of the imbalance in the observed sample. That is, during the induction of a discriminative model, we are implicitly estimating properties of P and G. In the case of empirical error minimizing linear separators, we are effectively estimating the density of points around the intersecting tails of the distributions. Two-sample bootstrapping provides a direct mechanism for estimating this boundary. In general, bagging will improve classifier performance when the individual members comprising the ensemble are high-variance – this is exactly the case with classifiers induced from different undersampled training datasets.

2.2.4 Simulations

We performed simulation experiments to systematically explore the empirical implications of the preceding sections. We constructed a simple generative model for creating instances that allowed us to experiment with various factors that, given our theoretical assumptions and above derivations, ought to influence the relative performance of various techniques for handling imbalance. The objective here is to use this simple model to elucidate the conditions under which different undersampling techniques might be effective.

We are specifically interested in exploring the scenarios in which undersampling, and/or bagging classifiers induced over undersampled datasets (hereupon referred to simply as bagging) outperforms other strategies for learning under imbalance. In particular, we consider SMOTE (35), and cost weighted-SVM. Obviously, there are many other existing techniques for handling imbalance with which we could have experimented, but the selected approaches are: 1) commonly used and 2) prototypical, as other techniques tend to be special cases or hybrids of these. In what follows we describe a simulation framework that allows us to explore in what circumstances different undersampling techniques work and when they will not.

2.2.4.1 Simulation Framework

The generative model we used in our simulations is described by the plate diagram in Figure 2.5 (note that all variables denoted by π parameterize Bernoulli distributions). This is essentially Naïve Bayes de-constructed, in the sense that when generating an instance x, feature-values are drawn conditioned on the instance label but independent of one another. In this simulation, we consider only binary features, i.e., each feature is either 1 or 0. We will denote feature j of instance i by x_{ij}. Without considering additional dataset characteristics (sparsity, noisiness), the label y_i, is 1 (i.e., a minority instance) with probability π^y. We associate with each (non-noisy) feature a *polarity*, which is drawn from a Normal distribution. The mean, μ, of this distribution determines how strongly features correlate with their labels – thus a larger μ here implies an 'easier' task.[1] The generative procedure we used is as follows.

First, each feature j is designated as a noisy feature with probability π^N (i.i.d.). For each instance x_i, we set $y_i = 1$ with probability π^y, else $y_i = 0$. We then generate feature-values for x_i conditioned on y_i and the 'feature-level' variables p^S and p^N. In particular, x_{ij} is set to 1 with a probability reflecting either polarity, noise (if j has been designated as a noisy feature), or sparsity. In the first case (noise), the probability of x_{ij} assuming 1 is governed by a polarity drawn from a (truncated) normal with mean μ. If, instead, the feature has been designated as noisy then x_{ij} is selected as 1 or 0 randomly, independent of y_i. Finally, every x_{ij} is zeroed out with probability π^S, inducing the desired sparsity.

Aside from the prevalence π^y, there are two parameters of particular interest that affect feature values: π^N, which dictates the amount of feature-noise, and π^S, which we call 'sparsity'. The former encodes the expected proportion of features that will contain no information regarding the label of the instances in which they are observed, i.e., features for which it is the case that $p(x_{ij} = 1|y_i) = p(x_{ij})$; no polarities are associated with such features. The latter (sparsity) encodes how sparse the generated instances will be: we zero out any given feature in any instance (independent of its label) with probability $1 - \pi^S$.

[1] We truncate this Gaussian to constrain values between 0 and 1, however we note that the μ's we use are near enough .5 (and the variances we use small enough) that such values are extremely unlikely in the first place.

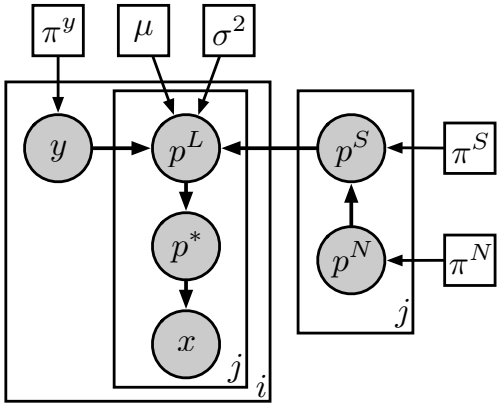


Figure 2.5: The plate diagram corresponding to our simulation scenario. See text for details.

To summarize, we use a simple multinomial model to generate data. This generative model allows us to explore the effects of various parameters of interest, including: dimensionality (d, which is external to the plate diagram, as it is set prior to generation); the degree of imbalance in the dataset π^y; the proportion of uninformative features in the data π^N; and the sparsity of the data, π^S. The final probability of a feature j being set to 1 in a generated instance is governed by $p*$ in Figure 2.5; this may either reflect the (truncated) polarity of the feature, or chance (if j has been designated 'noisy'), or it may have been zeroed out due to p^S.

2.2.4.2 Results From Simulation Experiments

We now present a series of experiments that explore the effects of altering the parameters outlined above to support the arguments presented in earlier sections. In all experiments shown, we fixed μ at .6, with a relatively tight σ^2 of .02 – i.e., the results shown are for datasets in which non-noisy features are relatively strongly correlated with classes. In all experiments, we generated both a training and a test set with the same parameters.

SMOTE requires specifying the percentage of synthetic minorities to be added to the dataset; for example, setting this to 100% will effectively double the minority class size by adding synthetic instances. For our purposes, we ran experiments with this parameter set across a few orders of magnitude (100% and

1000%) and display results for the best performing parameter.[1] For **weighted-SVM**, we used LibSVM's (34) implementation, and similarly experimented with a few orders of magnitude (100, 1000) for the parameter expressing the cost of false negatives relative to false positives, again showing the best result achieved for each experiment.

For **undersampling**, we threw majority instances away at random until the training set was balanced. Finally, for **bagging**, we built an ensemble of eleven classifiers induced over independently constructed undersampled datasets,[2] and predictions were taken as a majority vote over these. Both undersampling and bagging therefore include a stochastic element. We thus performed 10 independent iterations of each experiment with these methods to assess variability (error bars in the plots show best and worst performance over these runs).

Classifier evaluation over imbalanced datasets is inherently tricky. In practice, the relative costs of false positives (negatives) would have to be somehow elicited from the domain expert, and a weighted metric reflecting these costs could then be used to assess performance. For this work, we don't have these costs explicitly, and thus we take the standard approach of using a weighted harmonic mean of recall and specificity. Specifically, we use F_2^{spec}, in which sensitivity (also called recall) is considered more important than specificity.[3] These metrics were defined in Equations 2.1 through 2.4.

The first experiment we conducted considered the effect of increasing dimensionality on the induced classifiers' performances. According to Section 2.2.1, increased dimensionality should lead to decreased utility of the empirical-cost adjustment strategies (SMOTE and cost weighted-SVM), because in general as dimensionality increases, so too will the likelihood of the training data being separable, modulo the prevalence π^y and training sample size $|\mathcal{D}|$. For these experiments, we set the sparsity parameter π^S to .5, and we did not include any noisy features, i.e., all features were informative. The results are shown in Figure 2.6. In all cases, SMOTE and weighted-SVM both improve performance rela-

[1] These almost universally performed the same.

[2] The committee size of eleven was arbitrarily selected.

[3] Recall that we use specificity rather than the more popular precision when calculating F_2 because these metrics together provide more information than recall and precision.

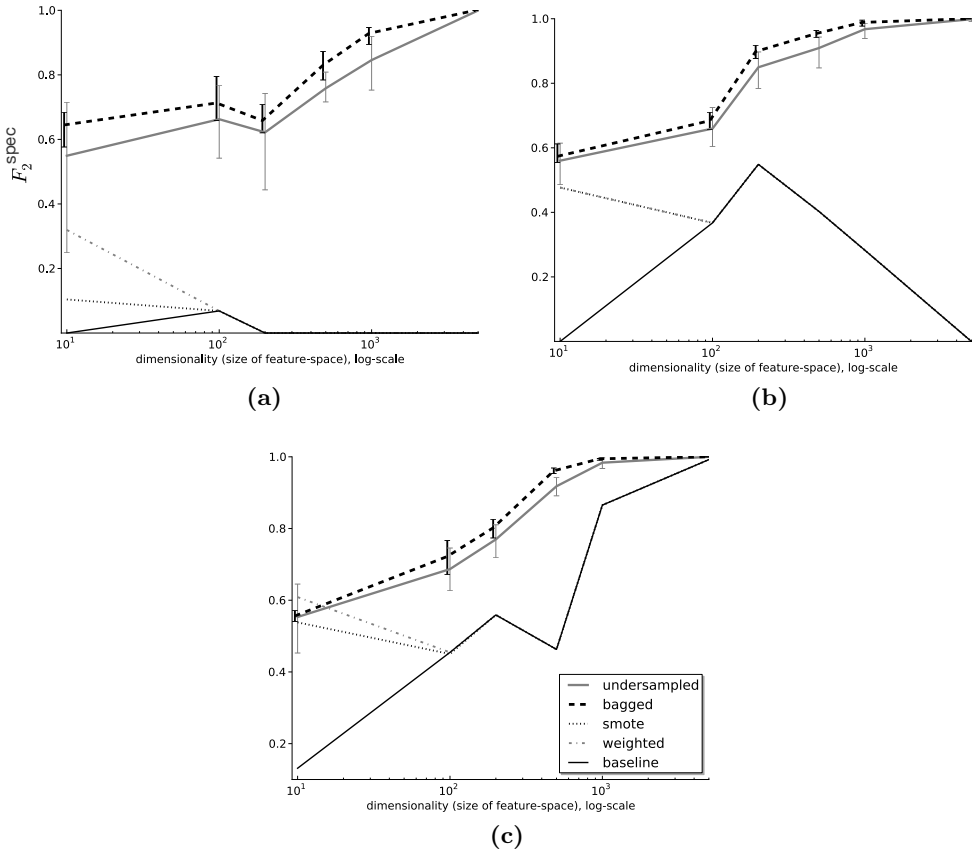
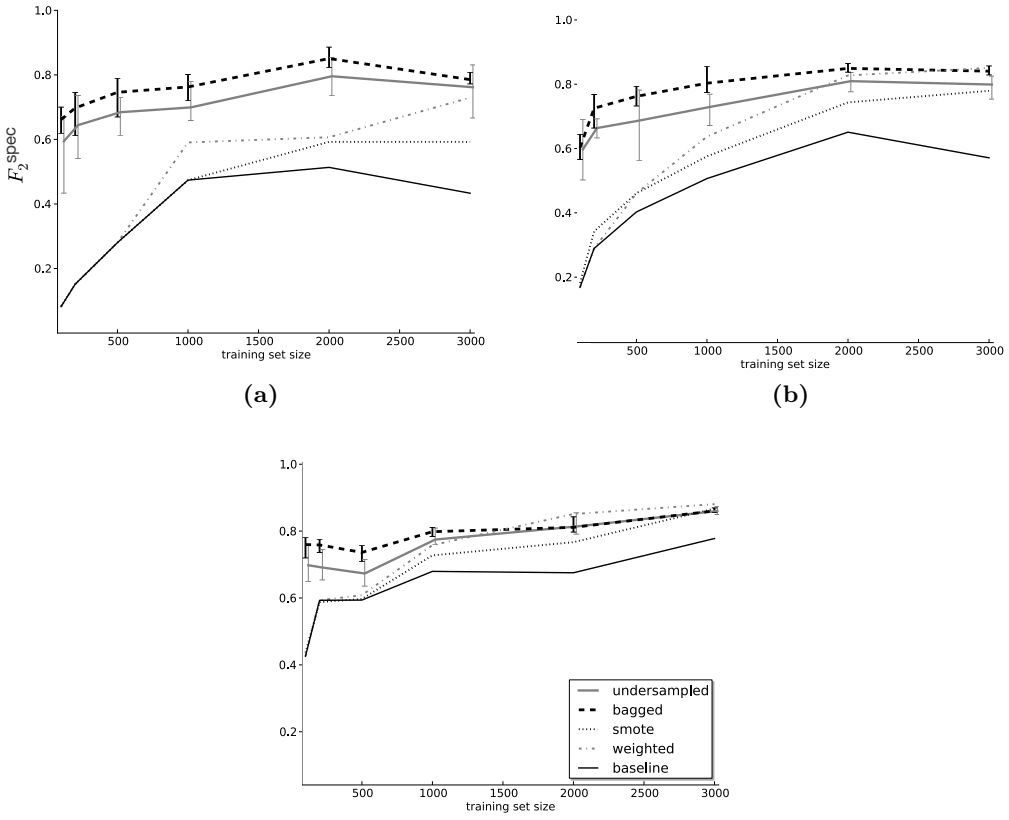


Figure 2.6: Simulation experiments investigating the relationship between dimensionality and F_2^{spec}. Dimensionality runs across the x-axis (log-scale) from 10 to 1000 dimensions. The plots show results for experiments with varying levels of minority prevalence π^y. In particular, sub-figures (a), (b) and (c) correspond to $\pi^y = .05$, $\pi^y = .1$, $\pi^y = .2$, respectively. For all experiments, the training and test set comprise 100 and 1000 examples, respectively. See text for further details.



(a) **(b)** **(c)**

Figure 2.7: Simulation experiments investigating the relationship between training set size and F_2^{spec}. In all experiments, the dimensionality of the feature space is fixed at 100. The minority prevalences in sub-figures (a), (b) and (c) correspond to $\pi^y = .05$, $\pi^y = .1$, $\pi^y = .2$, respectively.

tive to baseline SVM at lower dimensionalities, but their relative performance regresses to the baseline as the dimensionality increases, as we predicted.

The 'hump' seen in the first two cases appears because up to a certain dimensionality, the additional informative features increase model recall. However because of the low prevalence, the classifier is unable to learn which are the features associated with the minority class in higher dimensions. In 2.6(c), the prevalence appears to be sufficiently high to ameliorate this issue. Note that bagging not only reduces variance with respect to only undersampling (as can be seen in the corresponding error bars), but also performs better, on average.

The second experiment shown examines the relationship between the training set size ($|\mathcal{D}|$) and classifier performance. Figure 2.7 plots this against performance, again for three prevalences (.05, .1 and .2). In the first case, when the prevalence is low, the undersampled and bagged approaches consistently dominate, until the training set size reaches $\sim$3000, at which point weighted-SVM manages to catch up. When the degree of imbalance is less extreme, e.g., .1 and .2, the empirical weighted cost methods more quickly achieve performance comparable to the sampling strategies. This is precisely what we would expect in light of Equation 2.14. Note that undersampling and bagging again dominate, and again the latter both performs better and reduces variance, compared to undersampling alone.

We also considered the relationship between empirical error on the training set and the performance of the induced classifier. Our hypothesis was that empirical weighted cost strategies (e.g., weighted-SVM and SMOTE) would be effective in countering imbalance only when the baseline SVM incurred empirical error on the training set. This hypothesis follows directly from the discussion in Section 2.2.2. Strategies that upweight the cost of, e.g., false negatives w.r.t. false positives will work only insofar as they may push the separating plane demarcating the minority space until it encompasses the minority instance in the training set nearest the majority class. Once this outlying minority instance is correctly classified, modifying empirical costs will have no effect. Nor will SMOTEing work in this case, because due to the interpolative method of point

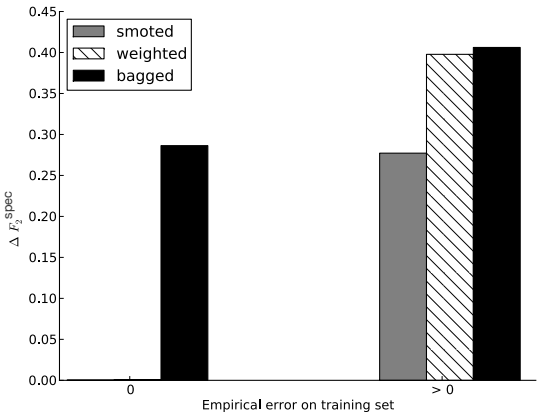


Figure 2.8: The y-axis is the average difference (improvement) in F_2^{spec} between the corresponding method and baseline SVM over hold-out test sets (ΔF_2^{spec} – when this difference is large, the corresponding method for handling imbalance was effective in that it improved performance). Three methods are shown: SMOTE, weighted-SVM and bagged. (Results for undersampled were similar to bagged). On the left-hand side of the plot, the average ΔF_2^{spec} is shown for the methods in datasets for which there was 0 empirical error (i.e., separable datasets). Note that the empirical weighted cost strategies provide no benefit over baseline, but bagging is effective. Results for cases where there was empirical error on the training set are shown on the right-hand side. In these cases, SMOTE and weighted-SVM are competitive with bagging.

generation, any synthetic point will necessarily fall inside of this outlying minority instance.

To explore this conjecture empirically, we ran experiments over fifty synthetic datasets generated at random. We drew the parameters dictating various properties of the dataset at random from sets of values we thought reasonable. In particular: for dimensionality, we drew uniformly from {10, 100, 500, 1000, 2000}, and for training set size from {100, 200, 300, 500, 1000}.[1] We drew π^N (noise) uniformly from {0, .2, .3, .4, .5, .6, .7, .8}, π^S (sparsity) uniformly from {0, .2, .5, .6, .7, .8, .95. 99}, μ (polarity) from {.55, .6, .65} and π^y (prevalence) from {.05, .1, .15, .2}.

Figure 2.8 displays summary results from these fifty datasets. In particular, we show the average improvement, in terms of F_2^{spec}, achieved by each of the strategies with respect to baseline SVM. The left-hand side corresponds to datasets over which baseline SVM achieved perfect accuracy, while the right-hand side plots results for datasets on which the baseline SVM incurred empirical

[1] In all cases we tested over a few thousand generated instances.

error. In the former case, neither SMOTEing nor weighting has any effect on classifier performance, but bagging does; when there is empirical error, however, both weighted SVM and SMOTEing are effective, thus supporting our conjecture. The message here is that bagging is effective even when a standard SVM can perfectly separate the training dataset, whereas empirical weighted cost strategies are not.

To summarize our results over synthetic datasets, we have shown that: 1) bagging/undersampling consistently outperformed other strategies in terms of predictive performance on synthetically generated imbalanced data (bagging doing so with lower variance than undersamping alone), and 2) as expected per our discussion in Section 2.2.1, undersampling/bagging was particularly effective, relative to SMOTE and weighted-SVM, in cases when the training dataset was not separable. Moreover, as is implied by Equation 2.14, this relative performance was observed to correlate with the prevalence (π) and the training set size ($\mathcal{D}$).

Strictly speaking, these results – and the theory developed above – hold only for linear separators. However, the fundamental problem does not change as a function of the classifier. The small sample from the P will inherently lead to decision surfaces that favor the majority class. This explains the empirically poor performance of non-linear classifiers induced on imbalanced datasets (80). We leave an analytic investigation into the non-linear case for future work.

2.2.5 Empirical Results on Benchmark Datasets

We now experiment with 'real' datasets to see if the patterns observed in the synthetic case (above) hold. We used sixteen datasets with varying degrees of imbalance; thirteen of these were taken from the UCI dataset repository, the other three from our biomedical text classification task. These are summarized in Table 2.1. Additional information regarding the latter three systematic review datasets is provided in Table 2.2: we will use these datasets throughout this thesis. Here we define the 'sparsity' of a dataset as one minus the expected

name	N	d	π	sparsity
car	1728	21	.040	.714
cmc	1473	24	.226	.625
ecoli	336	9	.104	.222
german	1000	61	.300	.721
glass	214	9	.079	0
haberman	306	3	.265	0
letter-a	20000	16	.039	0
letter-vowel	20000	16	.194	0
nursery	12960	27	.025	.704
pima	768	8	.349	0
splice	3190	287	.241	.790
vehicle	846	18	.251	0
yeast	1484	9	.289	.111
proton beam	4751	10025	.051	.993
copd	1600	6526	.122	.989
micronutrients	4010	11524	.064	.992

Table 2.1: Characteristics of the datasets we used in our experiments. The top thirteen are taken from the UCI dataset repository; the bottom three are systematic review datasets.

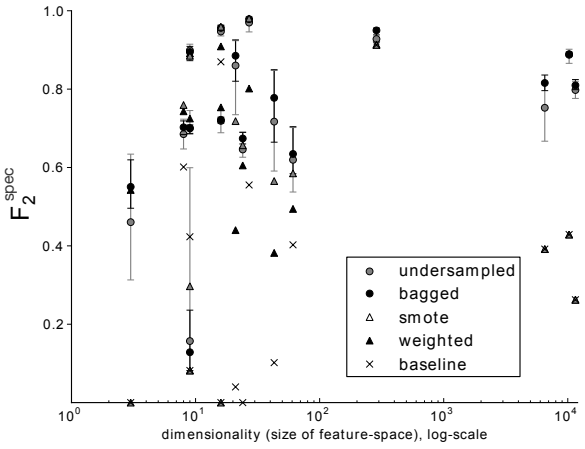


Figure 2.9: F_2^{spec} over test-sets for the datasets summarized in Table 2.1. Note that for the very high dimensional datasets, undersampling and bagging dominate (the latter again having lower variance).

Dataset	*Total citations (N)*	*Retrieved in full text (% of N)*	*Included in the systematic review (% of N)*
Proton beam	4,751	243 (5.1)	23 (0.5)
COPD	1,606	196 (12.2)	104 (6.5)
Micronutrients	4,010	258 (6.4)	139 (3.5)

Table 2.2: Three citation screening datasets that we will use throughout this thesis. We will usually use *level-1* decisions as labels as defined in the preceding chapter. That is, we will consider as *relevant* the citations that were retrieved in full text and *irrelevant* those that were not. The proton beam dataset is from a systematic review of comparative studies on charged particle radiotherapy versus alternate interventions for cancers (157). The COPD dataset is from a systematic review and meta-analysis of all genetic association studies in chronic obstructive pulmonary disease (32). The micronutrients dataset is from a systematic empirical appraisal of reporting of systematic reviews on associations of micronutrients and disease (38). Note the class imbalance in all three datasets.

proportion of features present in a given instance drawn from that dataset.[1] Sparsity is particularly relevant to textual data, wherein every word is relatively rare, and the vectors representing documents tend therefore to be sparse.

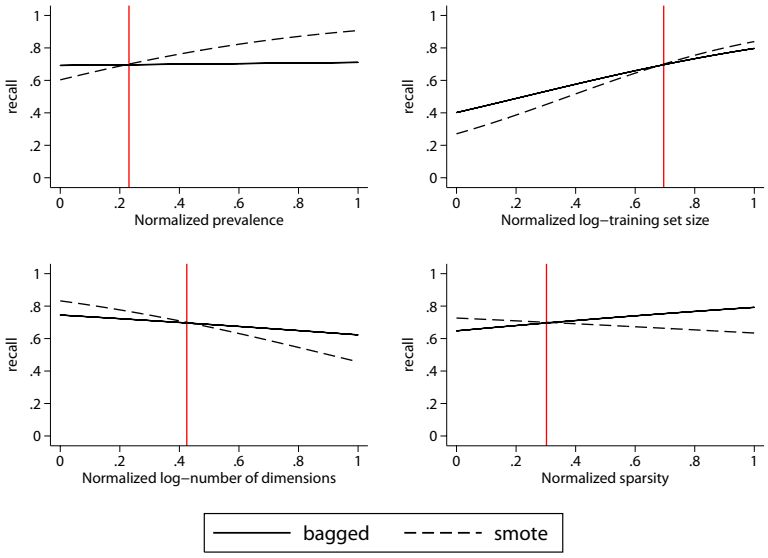
We first randomly split all of the datasets shown in Table 2.1 into train and test sets, comprising 10% and 90% of the corresponding datasets, respectively.[2] We then conducted the same experimental analysis as was described for the simulated data case in Section 2.2.4.

Figure 2.9 plots F_2^{spec} against dimensionality for all of the learners across all datasets. The most striking feature of this plot is the departure of bagging/undersampling at extreme dimensionalities: the difference in F_2^{spec} becomes substantial at dimensionalities of 10^4. At this point, as in our simulations, both SMOTE and weighted-SVM regress to baseline SVM. Another consistent pattern that emerges is that bagging again performs comparably (and often better) than undersampling alone and has a lower variance.

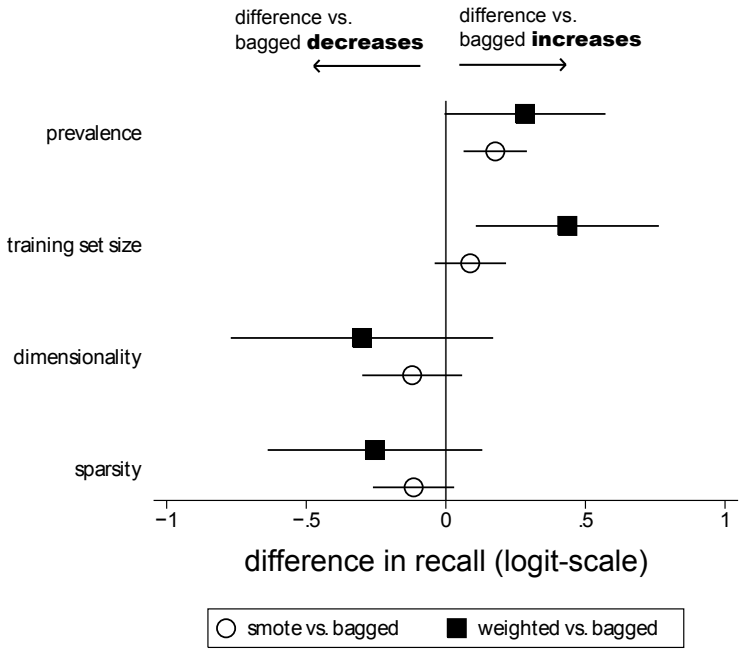
In addition to the utility of bagging for handling imbalance, the results here corroborate, and provide explanation for, previously reported observations. For example, Japkowicz (83) observed that as sample size increases, imbalance becomes less of a problem, in general. One can see why this is the case under our model: eventually a sufficient number of draws are made from the minority class, and it can thus be adequately characterized. (This is supported by Figure 2.7).

[1] We make no explicit distinction here between a feature not being observed (i.e., assuming a value of **0**) versus 'missing'.

[2] We use a relatively small portion of the data for training because in practice labeled data is typically scarce. In any case, we end up experimenting with a wide range of training set sizes due to the variance of the dataset sizes (N) in Table 2.1.



(a)



(b)

Figure 2.10: Results from a regression analysis of our empirical results. The top figure shows the estimated trends of the relative sensitivities of the bagging/undersampling and SMOTE methods. Specifically, each sub-plot shows the estimated effect of adjusting the corresponding parameter while holding all others constant at the point demarcated by the red lines. Bagging/undersampling works better than SMOTE, in terms of recall, as: prevalence decreases, the amount of training data decresases, dimensionality increases and as data becomes sparse. This can also be seen by considering the bottom plot, which shows the point estimates for the coefficients corresponding to these attributes.

Figure 2.9 is somewhat difficult to parse, and, further, it is restricted to one dimension (dimensionality), despite the fact that the other characteristics of interest (e.g., π) are not fixed, as they were in the simulations by construction. For interpretative purposes, we therefore performed an analysis on these empirical results to explore the effect of the different dataset characteristics on the respective methods for handling imbalance. More specifically, we evaluated the association between the recall of the five techniques and four characteristics of the datasets (prevalence, log-transformed training set size, log-transformed number of dimensions, and sparsity).

We used a two-level generalized linear mixed-effects regression that allows for between-classifier correlations within each dataset, and for common effects of the characteristics of interest across datasets. We are thus assuming that predictive sensitivity is a function of the particular dataset under consideration and the classification approach being used (these are the two levels): coefficients in the regression model thus correspond to the effect of the datasets and to that of the classifiers. In particular, we modeled the dependency of classifier performance (recall) on each characteristic with classifier-dataset interaction terms. Such hierarchical regressions are often used to explore which factors affect the relative performance of diagnostic tests (134).

Figure 2.10 displays the results from this analysis. Figure 2.10(a) shows how the predicted mean recall of SMOTE and bagging (i.e., predicted by the model induced over the empirical results) change for each classifier induced as a function of the dataset characteristics of interest. For each characteristic, we hold the values for the others constant; the vertical lines demarcate this fixed spot for each characteristic. The trends are as we expect: SMOTE works well when prevalence and training set size are large, but poorly when they are small, as is predicted by Equation 2.14. Similarly, SMOTE works comparatively well in lower dimensionality and sparsity (these can be seen as properties of the underlying distribution, P).

In Figure 2.10(b), we show the estimated coefficient of each of the aforementioned dataset characteristics in terms of their effect on the difference between the performance of bagged and that of the empirical weighted cost methods

(SMOTE and weighted-SVM). The circles and squares correspond to these point estimates for SMOTE versus bagged and weighted versus bagged, respectively, and the horizontal bars depict the 95% confidence interval. The directions of these coefficients are as expected, given our theoretical exposition and our simulation experiments; both SMOTE and weighted-SVM perform better (worse), w.r.t. the undersampled bagging approach, as the prevalence π and the training set size $\mathcal{D}$ increase (decrease). The reverse holds for dimensionality and sparsity: as these decrease, the effectiveness of the empirical weighted cost methods decreases relative to bagging. In the low-dimensional case, the empirical cost weighting strategies (SMOTE, weighted) are competitive with undersampling and bagging. In higher-dimensions, however, these strategies regress to the baseline.

2.2.6 Conclusions

In this section we have considered the task of classification in imbalanced scenarios from a probabilistic perspective. We ran simulation experiments that corroborated this framework. On this interpretation, we demonstrated the scenarios in which empirical error minimizing (linear) classifiers induced over imbalanced datasets will likely induce a biased separator. We also quantified the conditions when weighted empirical cost methods for mitigating the effects of imbalance, such as weighted-SVM, will likely fail to improve performance. We will next consider the problem of estimating class probabilities in imbalanced scenarios.

2.3 Probability Estimates for Imbalanced Data

Thus far we have considered the effect class imbalance has on classification. We will now shift our focus to estimating class probabilities in imbalanced scenarios. Obtaining good probability estimates is imperative for many applications, e.g., medical diagnosis, risk modeling, etc. The increased uncertainty (and typically asymmetric costs) surrounding rare events increases this need. Experts – and classification systems – often rely on probabilities to inform decisions. In our own case, for example, we would like to be cautious in classifying citations

as irrelevant; i.e., we would like only to do so when there is sufficiently high probability that this is indeed the case.

However, we demonstrate in this section that class probability estimates attained via supervised learning in imbalanced scenarios systemically underestimate the probabilities for minority class instances, despite ostensibly good overall calibration. Motivated by our exposition of this issue, we propose a simple, effective and theoretically motivated method to mitigate the bias of probability estimates for imbalanced data that bags estimators induced over balanced bootstrap samples. (This, of course, is similar to the above proposed ensemble method for classification). This approach improves performance on the minority instances without sacrificing overall calibration. We show that additional uncertainty can be measured via a Bayesian approach by considering posterior distributions over bagged probability estimates.

There has been a substantial amount of work investigating attaining probability estimates from supervised learning *in general*; notably (121) and (179). However there has been little work investigating the reliability of class membership *probability estimates* for imbalanced data. This is surprising because it is in such cases that probability estimates could potentially be of most use. In this work we focus on calibrated probability estimators, which transform raw scores from classifiers into probability estimates. We focus on calibrated methods because they are widely used, have desirable theoretical properties (42) and have been shown to achieve good probability estimation performance (121). Moreover, calibration is a general strategy that can be used to derive robust probability estimates from any classifier that provides a 'raw' output measuring confidence.

In this section we demonstrate that calibrated probability estimators produce systemically biased estimates in imbalanced scenarios. Specifically, while such estimators tend to have good overall calibration, they fare poorly in terms of their probability estimates for minority instances; these tend to be severely underestimated (i.e., low probabilities are wrongly assigned to truly positive minority examples). Such mistakes are especially problematic given the asymmetric costs common in imbalanced scenarios. Indeed, the motivation for attaining probability estimates is often to classify instances as belonging to the majority class only

when we are quite sure of it (this is the basic idea behind cost-sensitive learning (63)). But if the probability estimates for the minority instances are unreliable, then they are effectively useless for this purpose.

Aside from cost-sensitive learning applications, it is intuitively agreeable that a 'well-calibrated' probability estimator performs comparably with respect to both classes.[1] Consider that in a task with a minority prevalence of 1%, a model that uniformly predicts p=0% for *every* instance will be ostensibly well-calibrated (as we illustrate later) – according to most metrics and reliability diagrams (51) – despite its manifest uselessness. Another way of looking at the problem is to consider learning under the "covariate shift" assumption, in which the prevalence in the test distribution may diverge arbitrarily from that in the train set (22). The overall calibration of the aforementioned naive probability estimator would drop precipitously in cases in which minority prevalence is greater in the test than in the train set.

It is now well-appreciated that in classification tasks, accuracy is a poor measure of performance for imbalanced data (128, 170). Alternative metrics that emphasize good performance with respect to both classes are now widely accepted as more suitable for imbalanced data (e.g., sensitivity/specificity via ROC analysis, the G-mean) (80). This is analogous to measures of overall calibration being uninformative with respect to probability estimation in imbalanced scenarios. In Section 2.3.2, we propose a new metric for measuring probability calibration under imbalance: the *stratified Brier score*, which decomposes the classic Brier score (28) into elements reflecting its calibration w.r.t. minority and majority instances.

2.3.1 Estimating Probabilities in Supervised Learning

The standard method for estimating probabilities in the supervised learning framework is to regress measurements correlated with predicted class labels output by a trained classifier against the true target labels (121, 127). This process is called calibration. By convention these measurements are denoted by f_i, where i indexes instances. This calibration squashes the arbitrarily scaled f_i's into the

[1] We restrict ourselves to binary classification problems.

[0,1] range permissible for probabilities. When the sigmoid form is used, this method is referred to as Platt scaling (127). Platt scaling thus assumes that probabilities are generated as follows:

$$P(y_i = 1|f_i) = \frac{1}{1 + exp\{-\beta_0 - \beta_1 f_i\}} \tag{2.16}$$

Where the f_i's are scalars that are predictive of class membership. We focus on two specific post-training calibration strategies: Platt calibration with SVMs and with boosted decision trees. We selected these methods because they have been shown to out-perform other supervised learning algorithms with respect to class probability estimation (120, 121).

In the case of SVMs, f_i is the signed distance of instance i from the hyperplane w, i.e., $f_i = w^T x_i$. This was the method originally proposed by Platt (127), and is now widely used (104). Niculescu-Mizil and Caruana, meanwhile, have proposed attaining probabilities via calibrated boosted decision trees (120). More precisely, recall that in boosting one induces a sequence of learners h_0, h_1, ... , h_k over different distributions of the training set. These are in turn associated with a set of weights α_0, α_1, ... , α_k reflecting the their estimated performance. A prediction is then taken as a function over these, i.e., as $sign(\sum_j \alpha_j h_j(x))$. The natural value for f_i is then the sum of the weighted class predictions over the ensemble, i.e., $\sum_j \alpha_j h_j(x)$.

2.3.2 Evaluation of Estimated Probabilities

We now consider how to evaluate an estimator's predictions. This is trickier than evaluating classifiers because in the case of classification one predicts instance labels, which for validation data are directly observed. By contrast, true class probabilities are unknown, even when we have access to labels. One typically uses the labels as a proxy in evaluation by assuming that positive (negative) instances should be assigned high (low) probabilities. Thus an estimator that predicts 1.0 for all positive instances and 0.0 for all negative instances would be perfectly calibrated.

Figure 2.11 displays the overall and stratified residual errors of probability

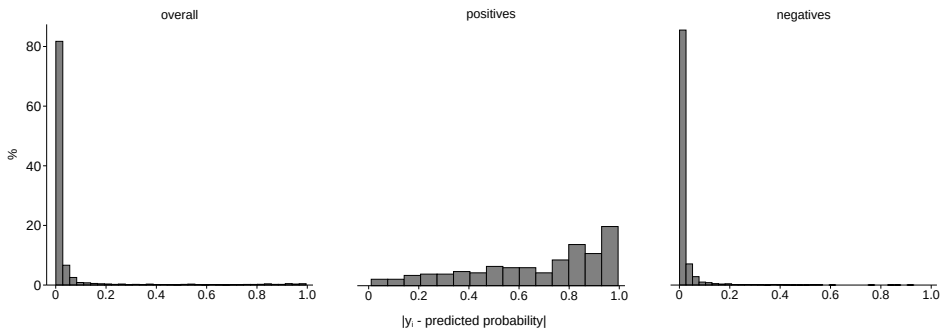


Figure 2.11: The bias of probability estimates attained via Platt regression for an imbalanced dataset. The x axis is the absolute difference between the observed labels and the corresponding probability estimates (i.e., $|y_i - \hat{P}\{y_i|x_i\}|$). Lower scores thus imply better calibration. Each plot is a histogram showing the densities of instances along this calibration metric. On the left, the histogram is shown for all instances; most instances are very near 0, implying good calibration. The middle and right-most plots show the corresponding histogram for the minority and majority classes, respectively. One can see that calibration is quite poor for the former class.

estimates (obtained via Platt's method) for the instances comprising a particular imbalanced dataset.[1] Specifically, each subplot shows histograms of the absolute differences between the true (observed) labels and corresponding probability estimates, i.e., $|y_i - \hat{P}\{y_i|x_i\}|$. Density to the left therefore suggests good calibration, as this implies probability estimates largely agree with the observed labels. For example, if $y_i = 1$ and $\hat{P}\{y_i|x_i\} = .99$, the difference would be .01. Were the estimate .01, on the other hand, the difference would be .99.

The left-hand side of Figure 2.11 shows this histogram for all instances, corresponding to overall calibration. Over 80% of instances are in the left-most bin, implying that the estimator is well-calibrated, i.e., its estimates do not much diverge from the observed labels. But this ostensibly good calibration belies the unreliability of the probability estimates for the minority instances. One can see this by looking at the middle plot, which includes only minority instances. In this case, the estimates diverge strikingly from the observed labels; indeed the model assigned a probability of belonging to the minority class of less than 20% to *most* of the minority instances. In other words, the probability estimates for instances comprising the minority class are completely unreliable (we demonstrate this on sixteen datasets in Section 2.3.6). Looking at the rightmost plot, which shows only the majority instances, one can see how this poor performance is hidden:

[1]The proton beam dataset described in the preceding section; see Table 2.2.

calibration is nearly perfect on the majority instances, and these dominate the dataset.

The *Brier score* (28) is one of the oldest and perhaps the most widely used metric for assessing calibration. Similar to the residual errors considered above, the Brier score measures the fit of probability estimates to the observed data. In particular, it is the average squared difference between the observed label and the estimated probability. Formally, this is defined in Equation 2.17 – here we are assuming that $y \in \{0, 1\}$, and we are denoting by N the size of the sample with which the model is being assessed (the test set).

$$\frac{\sum_{i=0}^{N}(y_i - \hat{P}\{y_i|x_i\})^2}{N} \tag{2.17}$$

Intuitively, this score is small when the probability estimates are near the true labels, and increases as they diverge. But there is a problem with the Brier score in the case of imbalanced datasets; calibration may be good *overall*, but poor for the rare class. Indeed, Figure 2.11 plots histograms reflecting each instance's contribution to the Brier-score. As we saw, this is low overall, but high for minority instances.

This phenomenon is analogous to the now well-appreciated observation that accuracy is a poor measure of classifier performance over imbalanced data (128). Consider that in a task with a minority prevalence of 1% a classifier that naively predicts that *every* instance belongs to the majority class will achieve 99% accuracy. Similarly in the case of probability estimation, a model that predicts p=0% for every instance will look to be well-calibrated, according to most metrics, despite its manifest uselessness. In the case of classification, alternative metrics that emphasize good performance with respect to both classes are now widely accepted as more suitable for imbalanced data (e.g., the F-score, G-mean). But the corresponding problem for probability estimation – good overall calibration masking unreliable estimates for minority instances – has not been addressed. We propose the modified Brier-score, which is more appropriate for assessing calibration in imbalanced scenarios.

$$BS^{+} = \frac{\sum_{y_i=1}(y_i - \hat{P}\{y_i|x_i\})^2}{N_{pos}} \tag{2.18}$$

$$BS^{-} = \frac{\sum_{y_i=0}(y_i - \hat{P}\{y_i|x_i\})^2}{N_{neg}} \tag{2.19}$$

Taken together, the Equations 2.18 and 2.19 provide much more information than the overall Brier score because they provide information regarding model calibration for instances drawn from both both classes. This is analogous to decomposing accuracy into sensitivity (recall) and specificity (or, similarly though not equivalently, precision).[1]

2.3.3 The Bias of Probability Estimates for Imbalanced Data

Recall that the common approaches to inducing probability estimates in supervised machine learning rely on post-calibration, i.e., fitting a (usually sigmoidal) function to raw outputs (39, 120, 121). Such strategies are theoretically motivated (39) and have been shown to produce good probability estimates (121).

As we will show in Section 2.3.6, however, post-calibration results in biased estimators. The problem of parameter estimation bias in imbalanced regression scenarios has been considered by the econometrics community, notably by King and Zeng (88), who demonstrate that $\hat{P}\{y = 1\}$ will be underestimated when $y = 1$ is a rare class.

The problem arises due simply to the model having observed more points drawn from the majority class (f_i's corresponding to majority instances) than from the minority. The model thus naturally fits the distribution generating the majority instances better than the distribution characterizing the minority instances. From this perspective, the reason for poor performance with respect to calibration is similar to the explanation for degraded classification performance discussed in the preceding section (166).

This intuition is best communicated graphically. Consider Figure 2.12, which depicts the fitted logistic for a simulated dataset. In this case, we assume the f_i's for each class are drawn from separate latent Gaussians for the two classes.

[1] Recall that precision is defined as $\frac{TP}{TP+FP}$, whereas specificity is $\frac{TN}{TN+FP}$.

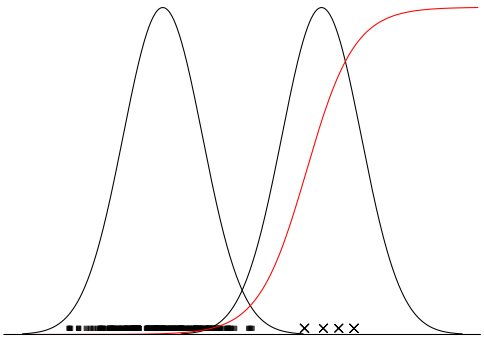


Figure 2.12: The bias inherent in fitting a logistic function to imbalanced data. Here we have two classes characterized by the shown latent Gaussian distributions. The points represent observed instances; for example the f_is of the majority (the ■s) and minority (the ×s) instances. Many fewer instances from the latter class have been observed. The red line is the shape of the logistic function fitted to the observed data $\hat{P}\{y_i|x_i\}$; it underestimates the conditional probabilities of minority instances belonging to the minority class.

(Note that this is in line with Hastie and Tibshirani's method of fitting f_i's from the respective classes to independent normals (76).) The fitted logistic is clearly biased with respect to the latent distributions; it is 'pushed over' toward the minority class, thus underestimating the conditional probability that $y = 1$ for minority instances.

More formally, based on results due to McCullagh and Nelder (114), King and Zheng (88) derive a closed-form expression for finite-sample size bias in logistic regression. Following their example for illustration purposes, consider a special case in which the true coefficient of the predictor is held constant (let $\beta_1 = 1$); then we need only estimate β_0. The predicted probability is:

$$\hat{p}_i = \frac{1}{(1 + exp\{-\hat{\beta}_0 - f_i\})} \tag{2.20}$$

King and Zheng (88) show that the expected bias in the estimate of β_0 is:

$$\mathbb{E}[\hat{\beta}_0 - \beta] \approx \frac{\tilde{\pi} - .5}{n\tilde{\pi}(1 - \tilde{\pi})} \tag{2.21}$$

Where $\tilde{\pi}$ denotes the true minority prevalence. In the case of imbalanced data, $\tilde{\pi} << 0.5$, implying that $\hat{\beta}_0$ will be an underestimate of β_0. This results in an underestimation of the probability that $y_i = 1$. Another intuition here is that the bias in the estimate is larger when imbalance is greater (because

$\lim_{\tilde{\pi}\to 0} \frac{1}{\tilde{\pi}(1-\tilde{\pi})} = \infty$).[1] This special case provides support for the empirical observation that imbalance is less of a problem when datasets are large: the bias in $\hat{\beta}_0$ is inversely proportional to the sample size (166).

These issues have long been recognized in the statistics and epidemiology literatures, and each community has developed distinct solutions to address the same problem. For example, there are several approaches that adjust for the small sample bias of logistic regression estimators (including the Haldane correction (169), Firth's penalized maximum likelihood (64), and Bayesian logistic regression). By contrast, epidemiologists have avoided such concerns altogether by using (in the majority of investigations) balanced case-control studies, wherein the number of cases (the 'positive' class) is set to be equal to the number of controls ('negative' class) by design (27). Thus the problem of bias due to imbalance in parameter inference is completely avoided in case-control studies due to their design.

2.3.4 Obtaining Better Probability Estimates for Imbalanced Data

From the above we can conclude that calibration in imbalanced scenarios will be biased, systemically affecting the conditional probability estimates for those instances comprising the minority class. This agrees with Figure 2.11 and our extensive empirical results (which we present in Section 2.3.6). What can be done to mitigate bias (and improve estimations) in imbalanced scenarios?

We propose undersampling as a means to accomplish this. Specifically, this entails discarding majority instances (at random) from the training set and calibrating probability estimates (e.g., estimating β) on this balanced set. This is analogous to the case-control sampling used in epidemiology that we mentioned above (27).

Intuitively, we can see the effect of undersampling on calibration by returning to the example introduced in Figure 2.12. The dotted line in Figure 2.13 shows the estimation when the logistic is fitted to a balanced sub-sample of the original dataset; contrast this with the solid line, which is the result of fitting the entire

[1] This limit is undefined in general; here $\tilde{\pi}$ is coming from the positive side.

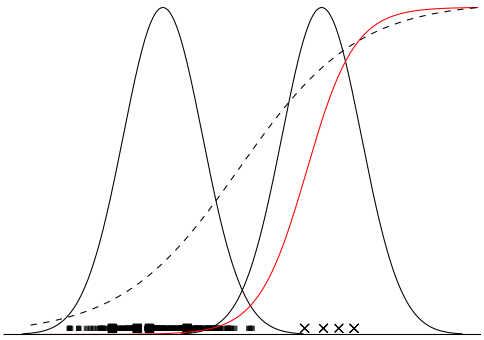


Figure 2.13: The effect of undersampling on fitting a logistic function to imbalanced data. The dotted line is the shape of the sigmoid induced fitting β only to the enlarged instances; the other ■s (majority instances) were discarded. For contrast, the solid red line is the corresponding sigmoid fitted to all data.

sample, and it is clear that the former mitigates bias. Theoretically, it is easy to see that the prevalence term in Equation 2.21 drops out (we note, however, that this is only applicable to the special case derived by King (88)).

2.3.5 Bagging Probability Estimates

While undersampling will mitigate bias, it also introduces randomness: the particular majority instances sampled will greatly affect the estimate $\hat{\beta}$. We can mitigate the variance inherent to this strategy via bagging (26). To bag probability estimates, we induce k calibrated models over corresponding balanced bootstrap samples. We then combine their outputs to form the estimated $\hat{P}\{y_i|x_i\}$ for a given x_i. The easiest method of combination is a simple average (Equation 2.22).

$$\hat{P}\{y_i|x_i\} = \frac{1}{k}\sum_{j=1}^{k}\hat{P}_j\{y_i|f_i\} \tag{2.22}$$

The simple average has the practical advantage of being easy to implement and fast to run. Indeed, despite building an ensemble of models, this approach may actually *reduce* running time. For example, inducing an SVM over an entire dataset often takes far longer than training several SVMs over small subsamples drawn from it.[1] That said, it is also natural to consider weighting the

[1]Recall that the training time of SVMs scales quadratically with the number of instances.

contribution of each constituent ensemble member by the certainty around their prediction, as in Equation 2.23:

$$\hat{P}\{y_i|x_i\} = \frac{1}{z}\sum_{j=1}^{k}\frac{1}{Var(\hat{P}_j\{y_i|f_{ij}\})}\hat{P}_j\{y_i|f_{ij}\} \tag{2.23}$$

Where $Var(\hat{P}_j\{y_i|f_{ij}\})$ denotes the variance of the prediction and z is the following normalization constant:

$$z = \sum_{j=1}^{k}\frac{1}{Var(\hat{P}_j\{y_i|f_{ij}\})} \tag{2.24}$$

It is natural to realize this weighting within the Bayesian framework by postulating a generative model and then sampling from the posterior probability estimates. Specifically, this can be done by assuming the following for each ensemble member j:

$$y_i \sim Bernoulli(p_i) \tag{2.25}$$

$$logit(p_i) = \beta_{0j} + \beta_{1j}f_{ij} \tag{2.26}$$

That is, for each model's predictive value f_{ij}, we assume that there exist coefficients that transform this value into the true probability. This is essentially the assumption made any time calibration is used. The estimate of the probability according to model j is then:

$$\hat{p}_{ij} = \frac{1}{1 + exp\{-\hat{\beta}_{0j} - \hat{\beta}_{1j}f_{ij}\}} \tag{2.27}$$

And, as before, we assume the true probability is an aggregate of the constituent estimates:

$$p_i = \frac{1}{k}\sum_{j=1}^{k}\frac{1}{1 + exp\{-\hat{\beta}_{0j} - \hat{\beta}_{1j}f_{ij}\}} \tag{2.28}$$

We fit this model using uninformative priors[1] over the βs. We thus sample from

[1]Recall that an uninformative prior states that we have no prior belief regarding a random variable.

the posterior of the β estimates. At the cost of added complexity, the Bayesian approach affords two major benefits. First, the uncertainty that each model has about its predictions is implicitly taken into account due to the sampling mechanism. A model uncertain regarding its β estimates will produce wide-ranging estimates during sampling, thus mitigating the contribution of their mean values. Conversely, a model with high certainty regarding its $\hat{\beta}$s will repeatedly make similar estimates, shifting the overall estimate (Equation 2.28) toward its mean.

The second benefit that the Bayesian framework provides is that of an additional measure of uncertainty. Specifically, one can take into the account the empirical posterior distribution over the aggregated probability estimate, which is not possible within the frequentist framework. We demonstrate the potential utility of exploiting this uncertainty in Section 2.3.7.

Note that taking a simple average (as in Equation 2.22) can be interpreted as ignoring the confidence in the estimates of the constituent members.

2.3.6 Empirical Results

The empirical results in this section demonstrate that standard supervised learning methods for probability estimation fare poorly on imbalanced data. More specifically, we analyze probability estimates attained via the two calibration methods reported to work best for supervised learning methods, namely (Platt-calibrated) SVMs and boosted decision trees (120, 121). We show that while overall calibration performance is good, calibration with respect to the minority class is often completely off.

We used the same sixteen imbalanced datasets described above (summarized in Table 2.1). We split each of these into train and test sets, the former comprising 10% of the dataset size (N). We induced probability estimators over the train sets using two distinct methods: SVMs and boosted decision-trees, obtaining probability estimates via calibration. These were selected due to their popularity and demonstrated performance in accurately estimating probabilities (120, 121). To measure the probability estimation performance we recorded

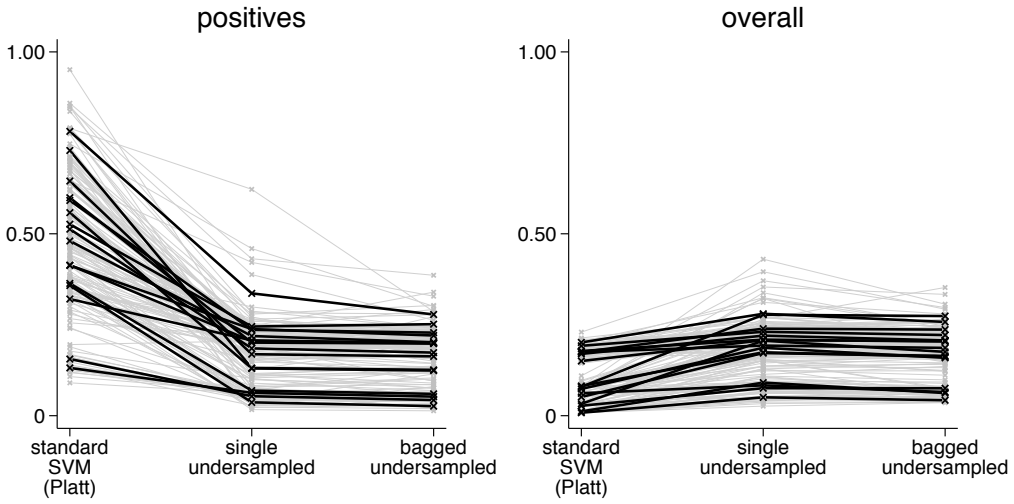


Figure 2.14: Calibration (Brier scores) for probabilities estimated using Platt calibrated SVM, undersampled and bagged/undersampled. The y-axis on the left-hand plot is the positive Brier score, which measures the goodness of the estimates for the minority class; on right-hand plot it is the overall Brier score. Recall that the Brier score measures the divergence of probability estimates from observed labels; lower scores are thus better. The standard method of estimating probabilities provides poor estimates for minority instances, but good overall calibration. Undersampling (and bagging) improves performance w.r.t. the minority class without sacrificing overall calibration.

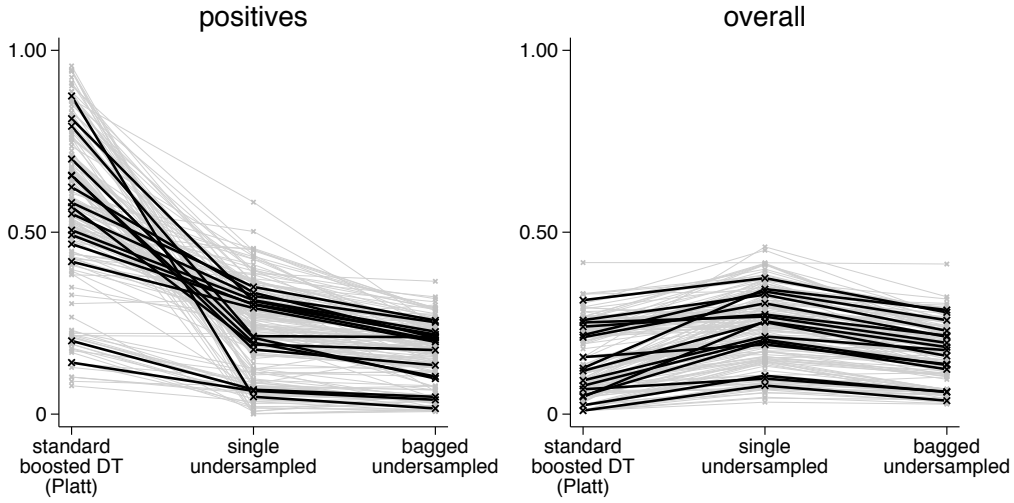


Figure 2.15: Boosted DT results (Brier scores of estimated probabilities). The results largely agree with those presented in Figure 2.14. In this case, bagging further improves calibration, in addition to reducing variance.

	standard SVM (Platt)			US & bagged		
dataset	BS (*SD*)	BS$^+$(*SD*)	BS$^-$(*SD*)	BS (*SD*)	BS$^+$(*SD*)	BS$^-$(*SD*)
car	0.033 (0.004)	0.729 (0.105)	0.004 (0.002)	0.175 (0.048)	0.126 (0.055)	0.177 (0.050)
cmc	0.175 (0.004)	0.592 (0.073)	0.053 (0.020)	0.237 (0.006)	0.228 (0.012)	0.239 (0.007)
ecoli	0.074 (0.013)	0.558 (0.178)	0.017 (0.011)	0.159 (0.029)	0.124 (0.032)	0.163 (0.033)
german	0.192 (0.006)	0.413 (0.067)	0.097 (0.027)	0.222 (0.003)	0.220 (0.016)	0.222 (0.008)
glass	0.079 (0.011)	0.782 (0.080)	0.019 (0.021)	0.274 (0.050)	0.278 (0.060)	0.273 (0.058)
haberman	0.201 (0.013)	0.526 (0.139)	0.082 (0.047)	0.258 (0.018)	0.252 (0.043)	0.260 (0.030)
letter-rec. *a*	0.008 (0.000)	0.156 (0.018)	0.002 (0.001)	0.042 (0.006)	0.043 (0.006)	0.042 (0.007)
letter-rec. *vowel*	0.150 (0.004)	0.599 (0.031)	0.042 (0.006)	0.205 (0.005)	0.198 (0.006)	0.207 (0.007)
nursery	0.012 (0.001)	0.358 (0.055)	0.003 (0.001)	0.062 (0.009)	0.026 (0.009)	0.063 (0.009)
pima	0.175 (0.009)	0.321 (0.069)	0.096 (0.030)	0.186 (0.010)	0.200 (0.020)	0.179 (0.015)
splice	0.062 (0.007)	0.131 (0.025)	0.040 (0.007)	0.065 (0.006)	0.052 (0.006)	0.069 (0.008)
vehicle	0.170 (0.008)	0.480 (0.078)	0.065 (0.029)	0.207 (0.015)	0.202 (0.040)	0.209 (0.023)
yeast	0.178 (0.012)	0.414 (0.078)	0.081 (0.021)	0.204 (0.009)	0.200 (0.013)	0.205 (0.013)
COPD	0.082 (0.009)	0.513 (0.107)	0.022 (0.010)	0.162 (0.038)	0.173 (0.032)	0.160 (0.042)
micronutrients	0.051 (0.002)	0.645 (0.067)	0.011 (0.003)	0.165 (0.017)	0.163 (0.027)	0.165 (0.019)
proton-beam	0.027 (0.001)	0.364 (0.052)	0.009 (0.003)	0.075 (0.010)	0.060 (0.014)	0.076 (0.011)

Table 2.3: Results (Brier scores) for Platt calibration via SVMs. The first three columns correspond to the overall, positive and negative Brier scores (Equations 2.17, 2.18 and 2.19), respectively. Standard deviations over ten independent runs for these are given within the parentheses. The last three columns show the same for the undersampled/bagged approach.

overall and stratified Brier scores (see Section 2.3.2). We repeated this procedure ten times to assess variance.

Figure 2.14 describes results over the datasets in Table 2.1 using three methods for estimating probabilities via SVMs: standard Platt, undersampled and the undersampled/bagging methods proposed in Section 2.3.4. The left and right sub-plots in Figure 2.14 correspond to the positive Brier score (BS^+, defined in Equation 2.18) and the overall Brier score (28) (Equation 2.17). Recall that we want to minimize the Brier-score. Each $\times$ represents a specific run: lines between these connect points generated from the same run, i.e., connect results for the three different methods on the same dataset, using the same test set. The black $\times$ are averages of the ten runs (lighter $\times$s are individual runs). The black lines depict the average difference in performance between methods on a given dataset (there are sixteen in all). These results are also summarized in Table 2.3, which shows the overall, minority and majority Brier scores for each dataset, and corresponding standard deviations of these scores over the ten runs.

The standard method of obtaining probabilities, Platt-calibrated SVMs, ostensibly slightly out-perform the undersampled strategies according to the overall Brier score reported in the right sub-plot of Figure 2.14. But BS^+ – the assessment of performance on the minority instances – tells a rather different story. Indeed, for *half* of the datasets the average BS^+ achieved using the standard method is greater than .5; in these cases, estimators calibrated using the standard procedure assigned, on average, a probability of $< .5$ to the proposition

	boosted DT (Platt)			US & bagged		
dataset	BS (*SD*)	BS^+(*SD*)	BS^-(*SD*)	BS (*SD*)	BS^+(*SD*)	BS^-(*SD*)
car	0.047 (0.014)	0.655 (0.175)	0.022 (0.013)	0.184 (0.032)	0.104 (0.081)	0.187 (0.033)
cmc	0.217 (0.013)	0.551 (0.034)	0.120 (0.022)	0.258 (0.018)	0.253 (0.018)	0.260 (0.023)
ecoli	0.093 (0.012)	0.570 (0.201)	0.036 (0.025)	0.176 (0.041)	0.176 (0.078)	0.176 (0.054)
german	0.258 (0.020)	0.506 (0.049)	0.151 (0.036)	0.229 (0.021)	0.227 (0.035)	0.230 (0.035)
glass	0.126 (0.052)	0.791 (0.113)	0.070 (0.061)	0.286 (0.031)	0.211 (0.042)	0.293 (0.036)
haberman	0.313 (0.046)	0.582 (0.174)	0.213 (0.104)	0.280 (0.050)	0.257 (0.059)	0.287 (0.078)
letter-rec. *a*	0.010 (0.000)	0.201 (0.019)	0.002 (0.000)	0.037 (0.006)	0.047 (0.027)	0.037 (0.007)
letter-rec. *vowel*	0.157 (0.001)	0.657 (0.015)	0.036 (0.004)	0.137 (0.008)	0.134 (0.010)	0.138 (0.010)
nursery	0.024 (0.002)	0.874 (0.109)	0.002 (0.002)	0.060 (0.007)	0.016 (0.008)	0.062 (0.007)
pima	0.252 (0.031)	0.419 (0.111)	0.161 (0.046)	0.196 (0.014)	0.199 (0.048)	0.194 (0.039)
splice	0.068 (0.014)	0.142 (0.053)	0.045 (0.014)	0.062 (0.005)	0.039 (0.005)	0.069 (0.007)
vehicle	0.211 (0.019)	0.493 (0.082)	0.117 (0.035)	0.213 (0.018)	0.204 (0.040)	0.216 (0.034)
yeast	0.240 (0.020)	0.467 (0.045)	0.148 (0.035)	0.216 (0.013)	0.219 (0.033)	0.214 (0.030)
COPD	0.119 (0.014)	0.624 (0.123)	0.049 (0.019)	0.160 (0.024)	0.209 (0.036)	0.153 (0.029)
micronutrients	0.078 (0.006)	0.812 (0.075)	0.027 (0.009)	0.135 (0.023)	0.215 (0.052)	0.129 (0.027)
proton-beam	0.050 (0.007)	0.701 (0.098)	0.015 (0.007)	0.123 (0.027)	0.098 (0.047)	0.125 (0.030)

Table 2.4: Boosted DT results. Note that the results for US & bagged differ from those presented in Table 2.3 because these are averages over a different set of randomized train/test splits.

that minority instances indeed belong to the minority class. Thus while overall estimation is good, the probabilities estimated for the truly positive (minority) instances are completely unreliable.

As the plot shows, undersampling prior to calibration sharply mitigates this issue. The BS^+ has a clear downward trend. In other words, undersampling substantially increases the quality of the probability estimates for minority instances (average decrease in the positive Brier-score of .315). Furthermore, while affected, this undersampling does not greatly sacrifice the overall calibration (average increase in overall Brier-score of .055). Moreover, bagging undersampled estimators fares even better; we see a similar decrease in the BS^+ with a lower hit in overall calibration. And, as expected, bagging reduces the variance of the Brier-scores. This can also be seen by inspecting the thin grey lines in Figure 2.14 which represent individual runs; the range, for example, is tighter in the bagged case, compared to undersampling.

The results for boosted decision trees (Table 2.4) tell a similar story. As in Figure 2.14, the left and right sub-plots in Figure 2.15 correspond to the positive and overall Brier scores, respectively. As in the case of SVMs, we observe that the probability estimation method proposed for boosted decision trees in (120) is well-calibrated, overall, but provides unreliable estimates for minority instances. The undersampling and bagging methods that we have proposed in Section 2.3.4 improve the probability estimates for minority instances while maintaining good overall calibration. In this case, bagging estimators improves

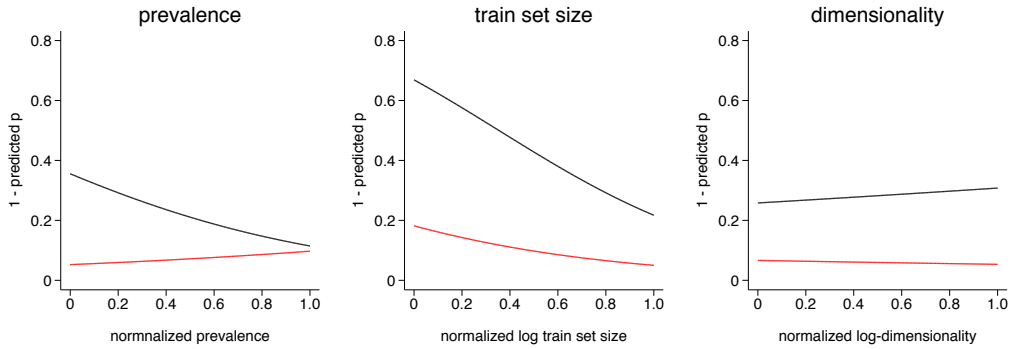


Figure 2.16: Fitted values from the linear mixed effects model. The predicted value of 1-$\hat{p}$ among the positive class for undersampled and bagged (red lines) and non-undersampled (black lines) is shown over different levels of prevalence (left panel), train set size (middle panel) and dimensionality (right panel). For each graph the level of the other factors was set at the mean value across the experimental datasets (e.g., when graphing the effect of prevalence, we set the train set size and dimensionality to their respective mean values).

calibration in addition to reducing variance (compared to a single undersampled estimator).

To explore whether dataset characteristics affect calibration performance (w.r.t. BS^+) we evaluated associations between the logit transformed 1-$\hat{p}$ values, the two techniques of interest (undersampled/bagged and standard SVM), and three dataset characteristics: prevalence, training set size, and dimensionality. We again used a linear mixed effects model for the logit-transformed 1-$\hat{p}$ values (the transformation was chosen to improve model fit) that allows for different effects of undersampling and bagging by dataset and common effects of the characteristics of interest across datasets (129). We plot the fitted lines for standard Platt and undersampled/bagged, over normalized (to within [0,1]) dataset characteristics in Figure 2.16. The results show that the improvement in calibration performance from using the undersampled/bagged strategy is greater, compared to standard Platt, when prevalence is low. This is a statistically significant finding ($p < 0.001$), and is what we would expect due to Equation 2.21. Although training set size and dimensionality did not reach statistical significance, the fitted lines suggest that the former may still have an effect (i.e., a lot of training data probably mitigates bias).

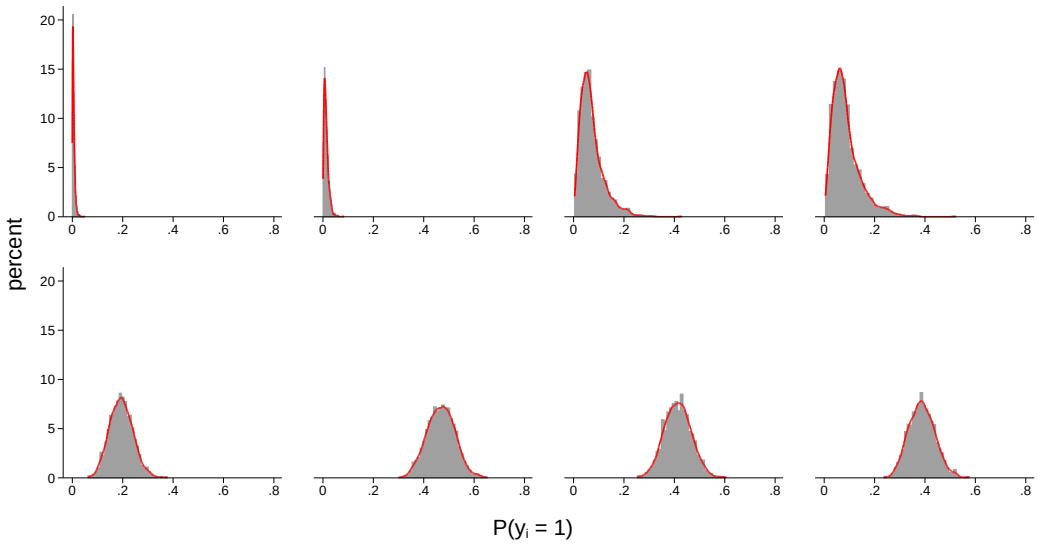


Figure 2.17: Empirical posterior distributions for four false negatives. The top row corresponds to this distribution for the standard model, the bottom to undersampled/bagged. See text for discussion.

2.3.7 Exploiting the Bayesian Framework for Additional Uncertainty

As discussed in Section 2.3.4, bagging probability estimators within the Bayesian framework affords a few advantages over the simple (frequentist) averaging approach. One such advantage is the ability to take into account an additional level of uncertainty, namely the empirical posterior distribution around the estimated $\hat{p}_i$s. For example, the median of this estimate may be shy of .5, but the 95% credibility interval (or any other specified interval) may encompass .5. For certain applications this may be useful information. More generally, the ability to marginalize over this posterior distribution could produce, for example, informed estimates of the true cost.

Figure 2.17 shows the posterior distributions around four instances from the COPD dataset (one per column) of minority instances assigned point estimates $< .5$; these would, presumably, be classified (wrongly) as negatives. The top row shows these posterior distributions using the standard (non-undersampled) model, the bottom for the undersampled/bagged model we have proposed. In the former case, the uncertainty is of no help: the estimates for each under-estimated minority instance are well below .5. The posterior distributions obtained via the bagged model, however, are potentially useful. Barring the left-most example, all

of these include .5, and indeed their mass hovers around it. One could easily exploit this additional uncertainty to be more cautious when making classification calls, or to obtain better estimates of expected costs.

2.4 Conclusions

We have provided a probabilistic interpretation of the effects class imbalance has on discriminative models and probability estimators. For the former case, we ran simulation experiments to corroborate this theory and demonstrated the scenarios in which empirical error minimizing (linear) classifiers induced over imbalanced datasets will likely induce a biased separator. Furthermore, we theoretically quantified the conditions when weighted empirical cost methods for mitigating the effects of imbalance, such as weighted-SVM and SMOTE,[1] will likely fail to improve performance.

It follows from the probabilistic interpretation of class imbalance developed in this chapter that re-sampling methods, specifically undersampling, should be applied in most imbalanced scenarios – in particular when prevalence is especially low or dimensionality is particularly high – as opposed to strategies that modify the objective function maximized during classifier induction to penalize false negatives more than false positives. Further, bagging should be used to reduce the variance of this approach. We motivated this advice theoretically and experimentally, and highlighted that this is in agreement with much of the prior experimental work investigating methods for handling imbalance.

In a similar vein, we have identified an analogous problem with supervised methods for estimating class probabilities in the case of imbalanced data: estimators systemically provide unreliable probability estimates for instances belonging to the minority class. We introduced a new metric, the stratified Brier score, to quantify this problem. We discussed the theoretical underpinnings of the issue and proposed a novel solution, namely inducing probability estimators over balanced bootstrap samples of the training data. We empirically demonstrated that

[1] We re-iterate that while SMOTE technically affects the training class distributions, it effectively behaves like a empirical cost weighted technique, due to its method for generating synthetic minority instances: see Section 2.2.2.1.

this simple approach mitigates the bias of the probability estimates, substantially improving the quality of the probability estimates for the minority class, without much sacrificing overall calibration. In short, better probability estimates can be had for imbalanced data by undersampling and bagging (specifically, we can improve calibration on minority instances a lot and suffer little for it in terms of overall calibration). Finally, we demonstrated that additional uncertainty can be exploited via a Bayesian approach by considering posterior distributions over bagged probability estimates.

3

Dual Supervision

In the preceding chapter, we addressed a problem common to real-world learning scenarios and inherent to the citation screening task: learning under class imbalance. We now turn our attention to making more efficient use of domain experts via novel forms of supervision. Specifically, in this chapter we look to exploit annotation beyond instance labels alone, namely in the form of *dual supervision*, in which domain expert(s) provide explicit information regarding features and their relationship to class labels. Exploiting such direct supervision is sometimes more efficient than learning from instance labels alone, and can thus mitigate the amount of labeling humans must provide, thereby reducing workload.

Consider the spam classification task alluded to in Chapter 1. In this task the aim is classify emails as *spam* or *not spam*. Given the prevalence of the 'Nigerian prince scam'[1] one knows that if the bigram 'Nigerian prince' appears in the text of an email, it increases the likelihood that the email is spam. Words and n-grams that correlate with specific classes are called *labeled features* (57). Intuitively, it makes little sense to expend effort learning this information indirectly via supervision at the instance level. Rather, we would like to allow the domain expert to impart this knowledge directly to the model. This is the aim of *dual supervision*, which refers to a family of methods that incorporate supervision from annotations on features in addition to instances during classifier induction.

Dual supervision is particularly attractive in the citation screening case reviewed in Chapter 1, as reviewers often have *a priori* knowledge regarding

[1] See `http://www.snopes.com/fraud/advancefee/nigeria.asp`

biomedical terms and their relationship to the clinical question at hand. For example, systematic reviews of drug efficacy often exclude trials that were performed on non-human animals (e.g., mice). Thus 'mice' and 'mouse' are good candidates for negative features, i.e., features that correlate with study exclusion. As a concrete example, in the case of the COPD systematic review (see Table 2.2), the expert indicated that 'allele' and 'COPD' were positive whereas 'mice' and 'cell lines' were negative. We will show that exploiting this type of supervision can indeed improve classifier performance, compared to learning from instance labels alone.

Empirical results achieved using models that exploit dual supervision are promising: methods that leverage labeled features consistently outperform those that learn from instance labels alone (116, 151, 182). Moreover, there is evidence that experts find labeling features less onerous than labeling instances. That is, acquiring feature labels is cheaper, in terms of labeling time, than acquiring instance labels (131, 144). Indeed, experts can often provide labeled features essentially for 'free', in cases that they are known *a priori*: they need only communicate them to the model (57).

In this chapter, we first review existing methods for learning under dual supervision. We then formulate a Support Vector Machine (SVM) variant that exploits labeled features: the *Constrained Weight-space SVM* (CW-SVM) (151). In addition to exploiting binary labeled features, the CW-SVM allows domain experts to provide *ranked* labeled features, and, more generally, to express arbitrary expected relationships between sets of features. We will later exploit the dual supervision paradigm in the context of active learning (Chapters 4 and 5).

While all of the work presented in this thesis is a product of close collaboration with colleagues, I would like to especially highlight Dr. Kevin Small's contributions to the content comprising this chapter. An earlier version of this chapter appeared in the *2011 Proceedings of the International Conference on Machine Learning* (ICML 2011).

3.1 Related Work

In this section we will review emerging methods for dually supervised learning. In Section 3.1.1 we discuss methods that extend the discriminative Support Vector Machine (SVM) framework. We then turn our attention to generative dually supervised approaches in Section 3.1.2.

3.1.1 Dually Supervised SVMs

One approach to dual supervision is the *annotator rationale* framework proposed by Zaiden et al. (181), in which experts highlight the features that influenced their categorization. This approach lends itself naturally to text classification, wherein the expert can highlight n-grams – i.e., words or phrases – responsible for their classification decision. For example, suppose the aim is to categorize movie reviews as 'positive' or 'negative'. Such sentiment analysis tasks are a natural fit for dually supervised approaches, and we will use the movies task throughout this thesis. Further suppose that the labeler is tasked to classify a scathing review of a new film. In addition to designating this review as 'negative', the annotator might highlight the sentence "This film was so terrible that I nearly walked out" as the *rationale* for their decision. These rationales may also be viewed as labeled features.[1]

Zaiden et al. (181) then exploit this information by constructing *contrast examples* for each provided rationale. Contrast examples are pseudo-instances that remove from labeled instances their associated rationales, i.e., the features (words) present in the rationale are zeroed out. The intuition is that the induced classifier ought to be less certain about these pseudo-instances than about the original examples that contain the rationales. This is expressed via *contrast constraints*, which specify that contrast examples should be some distance from their source instances – i.e., those containing the rationales. This distance can

[1]Strictly speaking, rationales are not equivalent to labeled features; they are arbitrarily long n-grams/sentences, rather than individual terms. In the quoted example, for instance, only 'terrible' would likely qualify as a labeled feature. Nonetheless, they can be used as (imprecise) labeled features.

be thought of as a margin. Denoting the linear separator by w, the contrast examples by v_{ij} (i.e., v_{ij} is instance x_i with explanation j removed), we have:

$$\forall i, j : y_i(wx_i - wv_{ij}) \geq \mu(1 - \xi_{ij}) \tag{3.1}$$

where μ is the size of the margin and we have introduced new slack variables ξ_{ij} for each contrast constraint. Next we add a third term to the objective function (Equation 1.2) reflecting our wish to satisfy these constraints. The C_2 parameter encodes how much emphasis is to be placed on satisfying the contrast constraints.

$$\frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{m} \xi_i - C_2 \sum_{i,j} (\xi_{ij}) \tag{3.2}$$

Zaidan et al.(181) demonstrated the efficacy of their approach with the movies dataset (123, 124), which comprises reviews of movies manually designated as positive or negative. Users were asked to categorize a training set of these reviews and provide their rationales by highlighting n-grams that explain their decision. The annotator rationale approach just described outperformed baseline SVM classification on the movies dataset. They also demonstrated that removing the rationales from the documents prior to training induced a model with poor accuracy, compared to an SVM induced over the documents with the rationales intact. This suggests that the human provided rationales are indeed important discriminative features.

Extending this work, Yessenalina et al.(178) proposed automatically generating rationales for the sentiment analysis case, achieving comparable performance.[1] Zaidan et al.(180) later re-cast the rationales approach within a generative framework, though the intuition remains the same.

Arora et al.(9) considered applying the annotator rationale framework in the context of structured features. Their approach exploits the part-of-speech structure in text to disambiguate individual tokens. They give the example of the following two statements: 1) "This camera has **good** features" and 2) "I did

[1]Note that this is similar to the standard labeled features approach; the best-performing generation strategy was a Polarity Lexicon, which is essentially a list of labeled terms for sentiment analysis.

a **good** month's worth of research before buying this camera". Clearly, the token **good** has different meanings in these contexts, and its presence in the second statement might puzzle a classifier. Subject-object relationships provide a means of disambiguating the token; Arora et al. (9) incorporated this information into the document representation. They achieved promising results; incorporating structural information outperforms the standard rationales approach (181).

An approach similar to rationales was proposed in earlier work by Sun et al. (153). In particular, they developed the *Explanation-Augmented SVM (EA-SVM)*. In this approach, as in rationales, an explanation specifies the features responsible for an instance's classification. To bias the learner toward parameter estimates that align with the provided explanations, Sun et al. (153) introduce constraints that encode the preference for explained examples to be further from the induced separator (more confidently classified) when they include their explanatory features than when they do not, similar to the contrast examples introduced above. They, too, demonstrate increased performance over baseline SVM on a few tasks.

We have reviewed several discriminative SVM-based dually supervised methods that achieve better performance than their standard instance-label only counterparts. In the next section we review generative approaches that incorporate dual supervision.

3.1.2 Generative Models for Dual Supervision

Several generative models for learning from dual supervision have been proposed. All of them essentially share the same underlying strategy of introducing bias into the model induction step to encourage parameter estimates that agree with the provided labeled feature information.

Melville et al. (115) proposed the *Pooling Multinomials* model as a framework for augmenting the standard naïve Bayes (NB) model with background knowledge in the form of labeled terms. This is accomplished by maintaining two separate conditional distribution tables; one for labeled features and another for unlabeled features estimated over the training data. We will denote these distributions by P_f and P_e, respectively.

The key idea is then to define a combined conditional distribution over words that combines P_f and P_e, that is, the *a priori* knowledge regarding the labeled features and the empirical parameter estimates. To calculate these *pooled* conditional probabilities $P(w_j|y')$ for the NB classifier:

$$\hat{y} = \arg\max_{y' \in \mathcal{Y}} P(y') \prod_j P(w_j|y') \tag{3.3}$$

where Y is the label set and w_j is word j. Melville et al. (115) considered both linear opinion pooling:

$$P(w_j|y') = \lambda_e P_e(w_j|y') + (1 - \lambda_e) P_f(w_j|y') \tag{3.4}$$

and logarithmic pooling:

$$P(w_j|y') = Z \cdot P_e(w_j|y')^{\lambda_e} \cdot P_f(w_j|y')^{1-\lambda_e} \tag{3.5}$$

The extent to which we trust or believe in the labeled features is encoded with the scalar parameter λ_e, which here is set using a sigmoid weighting scheme based on the error of the corresponding model (the labeled features model or the standard NB model) over the training set *a la* boosting.

Given a total vocabulary $\mathcal{V}$, a set of unlabeled features $\mathcal{U}$ and a set of labeled features $\mathcal{P}$ comprising positively and negatively labeled features, which we will denote by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively, the expert estimates are defined as:

$$P(w_+|+) = P(w_-|-) = [|\boldsymbol{\alpha}| + |\boldsymbol{\beta}|]^{-1} = |\mathcal{P}|^{-1} \tag{3.6}$$

$$P(w_+|-) = P(w_-|+) = [r(|\boldsymbol{\alpha}| + |\boldsymbol{\beta}|)]^{-1} = [r|\mathcal{P}|]^{-1} \tag{3.7}$$

and the estimates for weights associated with unlabeled terms in $\mathcal{U}$ are defined to reflect the relative prevalence of positive/negative terms. Specifically:

$$P(w_u|+) = \frac{|\boldsymbol{\beta}|(1 - 1/r)}{(|\mathcal{U}|)(|\mathcal{P}|)} \tag{3.8}$$

$$P(w_u|-) = \frac{|\boldsymbol{\alpha}|(1-1/r)}{(|\mathcal{U}|)(|\mathcal{P}|)} \tag{3.9}$$

where $r \geq 1$ is the *polarity level*, i.e., the relative increase in likelihood that a document is positive if it contains $p \in \boldsymbol{\alpha}$. Together, the above formulas define the P_f model that is pooled with the standard NB (P_e), as in Equations 3.4 and 3.5. Melville et al. achieve promising results with this approach, outperforming baseline strategies.

In a similar vein, Settles (144) recently proposed a simple extension to the naïve Bayes approach that assigns different priors to the conditional probabilities $P(w_j|y')$ for the labeled features. These encode the prior expectation regarding w_j's affinity for class y'. In particular, recall that one typically 'smooths' the observation counts (this avoids zero-probability estimates for cases in which a specific feature is never observed in a particular class). Typically a pseudo-count of 1 is added to each feature w_j class y' conditional observation count (this is known as Laplace smoothing). Settles (144) proposes adding $1+\kappa$ to the observation count of w_j given y' when w_j has been labeled as being correlated with y', where κ is a scalar encoding feature polarity ($\kappa = 0$ for unlabeled features). This simple approach has the advantage of lending itself naturally to the multi-class case, whereas pooling multinomials does not. Furthermore, Settles demonstrated that this approach often out-performs the pooling multinomials model, particularly when further augmented with semi-supervised techniques (144).

Elsewhere, Druck et al. (57) proposed using labeled features to constrain the induced model's predictions on unlabeled instances using the *generalized expectation* (GE) framework (112), described briefly below. In short, they add terms to their objective function that encode preferences for class labels on unlabeled instances reflecting the labeled term distribution therein. This can be operationalized by penalizing parameter estimates that diverge from our prior expectations (e.g., with respect to KL-divergence).

A *generalized expectation criteria (GEC)* is a formalism for encoding arbitrary *a priori* expectations directly into model parameter estimation (112). GEC

uses constraint functions $\tilde{f}$ that map the current expectation of a model to a scalar, which encodes a penalty for violating some preference on the induced distribution over labels. For example, this function may be a distance function Δ between the reference (i.e., expert-provided) and estimated distributions, as shown in Equation 3.10; KL-divergence is frequently used.[1]

$$G_{\tilde{f}}(E_\theta[f(x)]) = -\Delta(E_\theta[f(x)], \tilde{f}) \tag{3.10}$$

This scalar is then added to the parameter estimation objective function. GE terms can be viewed as a prior that is an arbitrary function over 'side-information' not directly available to the model. GEC may be used by themselves, or in conjunction with other terms. It can be shown (112) that GE is a general formulation which subsumes several popular parameter estimation methods as special cases (e.g., maximum likelihood).

GEC thus provides a means of incorporating *a priori* knowledge into parameter estimation. Using this machinery, Druck et al. (57) used GEC to exploit labeled features. In particular, they used GEC in conjunction with a discriminative probabilistic model parameterized by θ for text classification (they used a Markov random field, but any probabilistic model would do). Suppose, as before, that the expert provides sets of labeled terms $\mathcal{P}$. Then Druck et al. (57) add the constraint defined in Equation 3.11 for each $f \in \mathcal{P}$. Here $\hat{p}_f$ denotes the reference distribution for feature f. In the GE framework, experts specify the expected distribution of important terms over labels. For example, a user may expect that 80% of news articles containing the token 'basketball' will be sports rather than world news articles. Note that this is a finer-grained supervision than is provided by binary feature labels, and thus may prove too onerous for experts. To mitigate this problem, Druck et al. (57) provide a simple method of mapping binary labels to coarse distributions.[2] An estimate of the model parameters θ is then penalized if its predictions over unlabeled examples deviate

[1] To remain consistent with (57), here we define the objective function to be negative, thus assuming it will be maximized.

[2] Note that pooling multinomials performs a similar mapping of binary labels to conditional distributions.

from this expected distribution. This is formalized by the following objective function, where $x_f > 0$ implies feature f is present in x:

$$- \sum_{f \in \mathcal{P} \cup \mathcal{N}} D(\hat{p}_\theta(y|x_f > 0)||\tilde{p}_\theta(y|x_f > 0) - \sum_j \frac{\theta_j}{2\sigma^2} \tag{3.11}$$

Where $\hat{p}_\theta(y|x_f > 0)$ is the expert provided expected reference class distribution w.r.t. feature f and $\tilde{p}_\theta$ is the probabilistic model parameterized by θ that predicts a class for x. There are $|\mathcal{P}|$ GE terms in the objective function, one per labeled feature; these are defined to be the Kullback–Leibler distance between the reference and empirical (i.e., distribution of predictions with the currently induced model) class distributions for term f. Thus the model is penalized for choosing parameters that define a model that disagrees with the expert's prior expectations. The second term is a regularizer (specifically a non-informative Gaussian prior) that encourages the model to spread weights out over the parameters θ. This function is then optimized directly. Note that this approach does not consider any instance labels; a parameter vector is learned exclusively over labeled features.

We will show that the method we propose in the following section, the Constrained Weight-space SVM (CW-SVM), outperforms the methods reviewed above. Moreover, neither the generative nor discriminative models that have been proposed explicitly support *ranked* labeled features, or expected relationships between features. That is, most methods assume that experts provide binary labels on features that indicate if they are thought to be correlated with one class (or not). The GEC approach, meanwhile, allows for fine-grained specification of condition probability expectations of classes given features. In our view a middle-ground is more natural: experts may wish to rank features in terms of their correlations with a given class. The CW-SVM provides a means of exploiting such information.

3.2 The Constrained Weight Space SVM

In this section we present our novel formulation for exploiting expert-provided labeled features during classifier induction. Specifically, we extend the support

vector machine (SVM) model (45) by adding additional constraints to reflect this domain knowledge. Our method is unique compared to the existing approaches to learning with labeled features (reviewed above) in two ways. First, it provides a natural mechanism for directly encoding expert beliefs in the form of weight constraints. Second, our method is able to exploit *ranked* labeled features; e.g., in the case of sentiment analysis, *great* and *good* are both indicative of a *positive* movie review, the former is *more* indicative of this than the latter. The proton beam systematic review (157) provides another example: in this case the expert indicated that *hadrontherapy* is more indicative of a relevant abstract than *proton ion*, and conversely that *electron beam* is more indicative of an irrelevant abstract than *photon beam*. Such a ranking is natural in many domains, and as we shall see, exploiting this ranking can improve classifier performance over strategies that do not use rankings.

The remainder of this section is organized as follows. We first briefly review the standard C-SVM, which we build upon in Section 3.2.2 to realize the first two formulations of our proposed *constrained weight space SVM* (CW-SVM): the first supports only polar (binary) pairwise preference constraints, while the second allows for ranking of labeled features. We give a general formulation of the CW-SVM, of which the two aforementioned variants are special cases. We conclude the presentation of the CW-SVM, providing a concrete instantiation of the general case that allows an expert to encode knowledge regarding sets of labeled features and their polarity relative to one another. We report experimental results over a sentiment analysis task and two systematic review datasets from our motivating task in Section 3.2.3 – providing an empirical comparison with existing methods (just reviewed) that exploit dual supervision. Finally, we close the chapter with concluding remarks in Section 3.3.

3.2.1 Preliminaries

Recall that we are focused on learning binary linear classifiers of the form $f(\mathbf{x}) = sgn(\mathbf{w} \cdot \mathbf{x} + b)$ where $\mathbf{x} \in \{0,1\}^d$ is a d-dimensional feature vector representation of the item being classified, $\mathbf{w} \in \mathbb{R}^d$ is a d-dimensional weight vector, and $b \in \mathbb{R}$ is a learned threshold (i.e., bias element). Following conventional notation, let

$y \in \{-1, 1\}$ denote the label associated with an item. Given a set of m training instances $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, the goal is to inductively learn classifier parameters $\{\mathbf{w}, b\}$ that generalize well to unseen data.

We build upon the C parameterization for soft margin SVM classifiers (45). Defining $\boldsymbol{\xi} \in [0, \infty)^m$ as a slack variable vector to minimize instance-wise hinge loss and C as a tradeoff parameter between misclassification error and regularization, recall that the C-SVM (45) formulation is given by:

$$\underset{\mathbf{w},b,\xi}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \tag{3.12}$$

$$s.t. \quad y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \xi_i \;\; \forall i = 1 \ldots m \tag{3.13}$$

$$\xi_i \geq 0 \qquad\qquad \forall i = 1 \ldots m \tag{3.14}$$

3.2.2 Constraining the SVM Weight Space

As discussed above, a domain expert may know that particular feature values are correlated with one class or another. In the case of inducing a weight vector to discern positive from negative reviews, for example, we know *a priori* that the word *terrible* ought to have a lower weight than the word *terrific* (i.e., $\mathbf{w}_{terrific} > \mathbf{w}_{terrible}$). We augment C-SVMs to exploit such information by biasing the algorithm toward weight vectors in the hypothesis space that satisfy these constraints. More specifically, our method directly encodes expert knowledge regarding features through the definition of weight constraint sets, $p \in \mathcal{P}$, each comprising a set of binary relationships $\{\alpha, \beta\}_{\alpha,\beta\in p}$ that encode beliefs regarding the relative weight values (e.g., $\mathbf{w}_\alpha \geq \mathbf{w}_\beta$).

Generally, we call this model the constrained weight space SVM (CW-SVM). In the remainder of this section, we describe a sequence of CW-SVM instantiations. We begin with the relatively straightforward but powerful approach of allowing the expert to specify a single set of independent *pairwise constraints* (PWCs), as this is the simplest case. We then proceed by generalizing the CW-SVM framework, allowing for the incorporation of a specific set of *function-based constraints* (FBCs). It should be noted that in all of the proposed variants only a small number of features need to be labeled to achieve performance gains over baseline strategies (as will be seen in the empirical analysis), leaving the remain-

ing weights associated with unlabeled features unconstrained but influenced by their value in relation to the explicitly constrained weights.

3.2.2.1 Pairwise Parameter Constraints

The simplest instance of explicit rank feature-weight constraints are *pairwise constraints* (PWCs).[1] In this case, we assume only that the domain expert has specified pairs $\{\alpha, \beta\}$ of labeled features such that the weight associated with α should have greater value than the weight associated with β. Once such a pair is specified, a scaling parameter $\rho_{\alpha,\beta}$ is associated with each PWC such that the distance between the two weights (e.g., $\mathbf{w}_\alpha - \mathbf{w}_\beta$) is maximized in coordination with the existing C-SVM parameterization. Considering Figure 3.1, an example PWC is that $\mathbf{w}_{terrific} > \mathbf{w}_{lively}$. Note that while the "ordering"

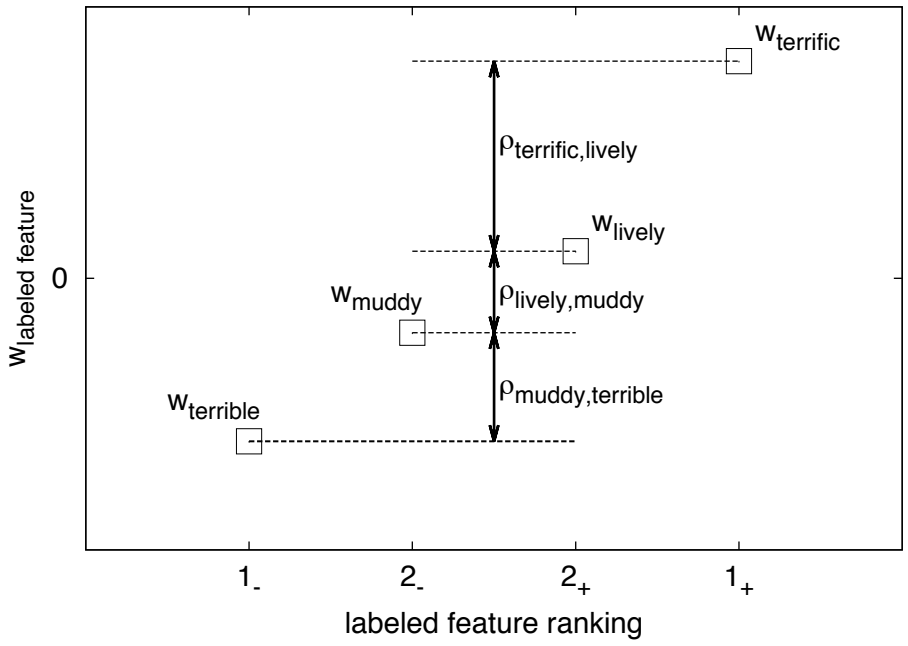


Figure 3.1: Weight bias induced by pairwise constraints.

of the weights is specified, the actual distance between weights, $\rho_{terrific,lively}$ is learned exclusively from the data. We now describe two specific CW-SVM formulations that exclusively utilize PWCs; *feature polarity* and *ranked features*.

[1] We cover the simple feature-polarity case in the next subsection, but begin here with rank constraints.

3.2.2.2 Feature Polarity

In the 'feature polarity' setting, we assume that the expert provides a set of positive labeled terms $\boldsymbol{\alpha}$ and a set of negative labeled terms $\boldsymbol{\beta}$. In this case, we generate $|\boldsymbol{\alpha}||\boldsymbol{\beta}|$ constraints and reward hypotheses where $\mathbf{w}_\alpha > \mathbf{w}_\beta$,[1] giving rise to the optimization:

$$\underset{\mathbf{w},b,\xi,\rho}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{m} \xi_i - C_2 \sum_{\alpha,\beta} \rho_{\alpha,\beta} \tag{3.15}$$

$$s.t. \quad y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i \;\; \forall i = 1 \ldots m$$

$$\mathbf{w}_\alpha - \mathbf{w}_\beta \geq \rho_{\alpha,\beta} \qquad \forall \alpha, \beta \tag{3.16}$$

$$\tau_- \leq \mathbf{w}_\alpha, \mathbf{w}_\beta \leq \tau_+ \qquad \forall \alpha, \beta$$

$$\xi_i \geq 0 \qquad \forall i = 1 \ldots m$$

Where τ 'boxes' the weight-space; we set these values by first fitting a standard SVM on the data and using the lower and upper bounds of the induced w as τ_- and τ_+, respectively. These 'box-constraints' prevent the optimization procedure from selecting weight-vectors with arbitrarily large ρ's (i.e., very large or small values corresponding to the terms for certain labeled features), which would be undesirable. It is true that the first term in the objective penalizes such behavior by looking to minimize $\mathbf{w}$. However, the amount of emphasis placed on this versus maximizing the ρ's will depend on the trade-offs encoded by the C terms. In our experience if one uses grid-search to set the C's, the problem of pushing the ρ's out ever further remains, and hence our inclusion of the box-constraints. Admittedly, however, this box-constraint solution is inelegant. An alternative formulation may constrain the ρ's to be positive, precluding the possibility that feature constraints are violated. This may mitigate the problem of very large ρ's, and is a direction we plan on exploring. In any case, the main point is that here we augment the C-SVM optimization problem by encoding a preference to separate the weights of features with known polarity, using the defined PWCs of Equation 3.16 and rewarding this separation in the objective function of Equation 3.15.

[1] Note that that this can be equivalently accomplished with PWCs that constrain positive (negative) feature weights to be greater (less) than the decision threshold.

3.2.2.3 Ranked Features

In the preceding section we described a method for incorporating labeled features with respect only to class polarity, which is similar to previous work on learning with labeled features (see Section 3.1). We now introduce machinery to exploit *ranked labeled features*. For example, while "terrific" and "lively" may be associated with a positive movie review and "muddy" and "terrible" with a negative review, an expert may want to specify that they believe $\mathbf{w}_{terrific} \geq \mathbf{w}_{lively} \geq \mathbf{w}_{muddy} \geq \mathbf{w}_{terrible}$. It is straightforward to derive a PWC formalism to include ranked features. Specifically, if we define $\alpha \succ \beta$ to indicate that $\mathbf{w}_\alpha \geq \mathbf{w}_\beta$ such that the rankings for α and β are adjacent, the following optimization problem captures ranked feature information:

$$\begin{aligned} \underset{\mathbf{w},b,\xi,\rho}{\text{argmin}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{m} \xi_i - C_2 \sum_{\alpha,\beta:\alpha\succ\beta} \rho_{\alpha,\beta} && (3.17) \\ s.t. \quad & y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i \quad \forall i = 1 \ldots m \\ & \mathbf{w}_\alpha - \mathbf{w}_\beta \geq \rho_{\alpha,\beta} \qquad \forall \alpha, \beta : \alpha \succ \beta && (3.18) \\ & \tau_- \leq \mathbf{w}_\alpha, \mathbf{w}_\beta \leq \tau_+ \qquad \forall \alpha, \beta \\ & \xi_i \geq 0 \qquad \forall i = 1 \ldots m \end{aligned}$$

Note that the "most weakly" positive labeled features are considered adjacent to the "most weakly" negative labeled features; in our above example $\mathbf{w}_{lively}$ would be considered adjacent to $\mathbf{w}_{muddy}$. We augment the C-SVM optimization problem in order to encourage separation between features with adjacent rankings using the pairwise weight constraints of Equation 3.18 and rewarding separation in the objective function of Equation 3.17. Note that the feature polarity formulation described in the previous section is a special case of ranked features where there are only two possible rankings.

3.2.2.4 CW-SVM: A General Formulation

As developed thus far, PWC formulations reward correct parameter "orderings" (with respect to *a priori* expert beliefs), but do not provide a means for encoding beliefs regarding the relative distances between the provided sets of ranked

weights. For some tasks experts may wish to express an intuition such as "The terms *horrible* and *awful* are exponentially more indicative that a movie review is negative than are the terms *convoluted* and *long*." We now present a general formulation of the CW-SVM that allows the expert to formally express his or her domain knowledge.

First, we define ranked feature sets where $r_p(x)$ denotes the expert defined rank associated with each labeled feature such that $r_p(x) > 0$ indicates a ranking associated with positive class labels and $r_p(x) < 0$ is associated with negative class labels. We encode the rankings numerically as follows: the terms belonging to the *most* positive set map to rank 1; terms in the second most positive set to rank 2, etc. The same holds for negatively ranked terms, only the values are negated to encode polarity. In our running example from Figure 3.1 $r_p(\text{lively}) = 2$, $r_p(\text{terrific}) = 1$, $r_p(\text{muddy}) = -1$ and $r_p(\text{terrible}) = -2$.

Next we define a function g_p over ranks $r(\alpha)$ and $r(\beta)$ to provide a scalar expressing the expected difference in weight values of their sets' respective members. For example, consider Figure 3.2, where we are shaping both the positive and negative ranked features with separate exponential functions.

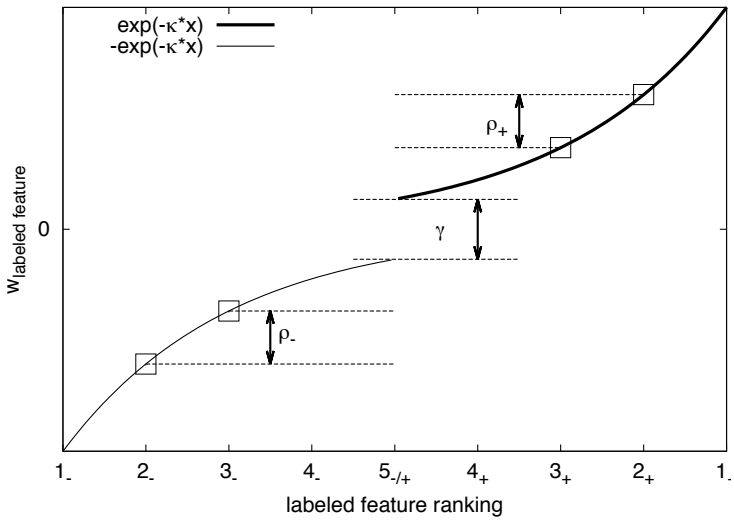


Figure 3.2: Weight space bias induced by function-based constraints

In this case, all of the weights associated with positively (negatively) ranked features are shaped along an exponential function where the distance between parameters is scaled by ρ_+ and ρ_-, respectively. In general, there can be many such functions for different sets of features, although this will likely be a small

number of functional families in practice (e.g., linear or exponential). Formally, we have the following general optimization problem:

$$\underset{\mathbf{w},b,\xi,\rho}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{m} \xi_i - \sum_{c \in \mathcal{C}} C_p \cdot \rho_p \tag{3.19}$$

$$s.t. \quad y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i \quad \forall i = 1 \ldots m$$

$$\mathbf{w}_\alpha - \mathbf{w}_\beta \geq \rho_p \cdot g_p(r(\alpha), r(\beta))$$

$$\forall \alpha, \beta, p : p \in \mathcal{P}; \alpha, \beta \in p \tag{3.20}$$

$$\tau_- \leq \mathbf{w}_\alpha, \mathbf{w}_\beta \leq \tau_+ \qquad \forall \alpha, \beta$$

$$\xi_i \geq 0 \qquad \forall i = 1 \ldots m$$

Provided with the feature constraint sets $\mathcal{P}$, the optimization procedure balances the minimization of the magnitude of $\mathbf{w}$ and the minimization of training error (as in C-SVM), while attempting to maximize the relative influence the constraint information through the scaling vector $\boldsymbol{\rho} \in \mathbb{R}^{|\mathcal{P}|}$ (the influence of these terms is influenced through their respective C parameters). Thus, while the expert-defined g_p determines the shape of the constraining function, the scale of the relative separation is still learned from data. The influence of the scaling parameters associated with each p is determined by the parameter C_p (which is set using expert knowledge or cross-validation over the training data). Once the quadratic program (QP) is specified, existing QP packages can be used to solve the optimization problem.[1] Using this formulation, an expert can define several sets of parameter constraints and functions that define beliefs about their relationships. In the next section, we describe a particular instantiation of the CW-SVM that includes *function-based constraints* (FBC).

3.2.2.5 Function-based Constraints

The PWC formulations of Section 3.2.2.1 are specific instantiations of the general CW-SVM in which there exists an independent function $g_p(r(\alpha), r(\beta)) = 1$ for each pairwise constraint (i.e., there is one parameter constraint in each parameter constraint set). However, there are situations where the expert may wish to provide the classifier information such as "$\mathbf{w}_{terrific}$ is much more posi-

[1] We use `CVXOPT` (48).

tive than $\mathbf{w}_{good}$ while $\mathbf{w}_{good}$ is slightly more positive than $\mathbf{w}_{lively}$." This is shown in Figure 3.2, where the aforementioned weights are biased to fit along the function $f(w) = e^{-\kappa \cdot r(\mathbf{w})}$ (where κ is a constant). In this case, we would define $g(r(\alpha), r(\beta)) = e^{-\kappa \cdot r(\alpha)} - e^{-\kappa \cdot r(\beta)}$ and constrain all of the positive ranked parameters to the shape of this function (therefore learning the scaling parameter ρ associated with a specified g). Here the expert would group these parameter constraints into a parameter constraint set p and specify a function to express relationships between pairs of labeled features in this set. By allowing the expert to specify this additional information, and thus inducing a stronger bias on the parameter space than PWC, we can further reduce the labeled data requirements, as demonstrated by our empirical results.

We now introduce a particular instantiation of CW-SVM where all of the positively labeled features are used to generate one parameter constraint set (which are related to each other by a single shaping function) and all of the negatively labeled features are used to generate a second parameter constraint set (which are related to each other by a second single shaping function). Finally, we define PWC along the *polarity border* to enforce a notion of margin among the labeled features. This results in the following optimization problem:

$$\begin{aligned}
\underset{\mathbf{w},b,\xi,\rho}{\operatorname{argmin}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{m} \xi_i - C_2 \sum_{\substack{\max(r(\alpha)) \\ \min(r(\beta))}} \rho_{\alpha,\beta} \\
& -C_3 \cdot \rho_1 - C_4 \cdot \rho_2 \qquad (3.21) \\
s.t. \quad & y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i \quad \forall i = 1 \ldots m \\
& \mathbf{w}_\alpha - \mathbf{w}_\beta \geq \rho_{\alpha,\beta} \\
& \qquad \forall \alpha, \beta : \max(r(\alpha)), \min(r(\beta)) \qquad (3.22) \\
& \mathbf{w}_\alpha - \mathbf{w}_\beta \geq \rho_1 \cdot g_1(r(\alpha), r(\beta)) \\
& \qquad \forall \alpha, \beta : \alpha \succ \beta, r(\alpha) > 0, r(\beta) > 0 \qquad (3.23) \\
& \mathbf{w}_\beta - \mathbf{w}_\alpha \geq \rho_2 \cdot g_2(r(\beta), r(\alpha)) \\
& \qquad \forall \alpha, \beta : \alpha \succ \beta, r(\alpha) < 0, r(\beta) < 0 \qquad (3.24) \\
& \tau_- \leq \mathbf{w}_\alpha, \mathbf{w}_\beta \leq \tau_+ \qquad \forall \alpha, \beta \\
& \xi_i \geq 0 \qquad \forall i = 1, \ldots, m
\end{aligned}$$

This form is general because there are infinitely many possible shaping functions which can be used to define FBCs. However, realistically there only a small number of functional families are useful in practice (e.g., linear, exponential, sigmoidal, etc.) – making FBC formulations feasible for expert specification in the common cases.

3.2.3 Experimental Results

For our experimental evaluation, we use datasets from our motivating task of biomedical citation screening (see Chapter 1). We note that, as in the preceding chapter, we use *level-1* labels as the target concept here. We also conduct experiments with a movie reviews dataset (123). We compare the CW-SVM against appropriate baselines (i.e., without labeled features) and existing strategies that exploit labeled features, described next. We note that, to our knowledge, this is also the first empirical comparison of these methods for learning with labeled features.

3.2.3.1 Methods

We summarize below the methods to which we compare the CW-SVM. These methods were reviewed at length in Section 3.1; here we provide the main intuition and implementation details.

Annotator rationales. Zaidan et al. (181) proposed the *annotator rationale* framework as a means of incorporating annotator 'explanations' into the training algorithm. This is done by having the expert mark the text (features) that most influenced their labeling decision. To then exploit provided rationales, several *contrast examples* are generated for each instance, which intuitively are examples assumed to be negative due to the forced absence of a particular rationale. The SVM algorithm is correspondingly modified with *contrast constraints* to encourage the model to find weights that are consistent with the expert-provided rationales. For more details, see Section 3.1. We do not have rationales in the case of systematic reviews, because our approach requires a small set of labeled terms as opposed to rationales for each instance (which doctors are not anxious to supply when conducting reviews). We therefore do not compare against the rationales approach for the systematic reviews datasets. We do, however, compare the CW-SVM to the rationales approach over the sentiment analysis task, using the methodology described in (181).

Pooling multinomials. As described above, the pooling multinomials model (116) extends the standard naïve Bayes model for text classification. In particular, they compute posterior estimates of a document belonging to a given class using both the standard naïve Bayes model and a generative 'background' model that incorporates labeled features (terms), which they refer to as the *lexical* model. The basic strategy in deriving their lexical model is to assign probabilities to the labeled terms reflecting their polarity, or class association. For technical details of their lexical model, see (116).

The estimates of these two models (multinomials) are then linearly combined with weights reflecting the accuracy of the respective models (as estimated via cross-validation over the training data). In particular, each model m (naïve Bayes, lexical) has an associated weight α_m computed as follows: $\alpha_m = log\frac{1-err_m}{err_m}$. Because of our emphasis on recall in the citation screening

scenario (and evaluation via F_2), we modify their approach slightly for these datasets such that the two models are combined according a weighted error; in particular, we use $err_m = \frac{fpr_m + \beta fnr_m}{1+\beta}$, where fpr_m and fnr_m are the false positive and false negative rates, respectively.[1] This modification improved the performance of their method on the screening task datasets, compared to their proposed method, which effectively optimizes for accuracy (we did not incorporate this change for movies, wherein accuracy is the metric of interest).

GEC. We used the Mallet(111) implementation of the GEC framework described elsewhere (57) and reviewed in Section 3.1.

CW-SVM. For the CW-SVM, we compared results using the following variants:

- **Polarity** - The PWC formalism wherein weights associated with positively labeled terms are constrained to be greater than all weights associated with negatively labeled terms (Section 3.2.2.2).

- **Ranked** - The PWC formalism in which adjacently ranked features correspond to associated constraints in weight space (Section 3.2.2.3).

- **FBCs** - The formulation of Section 3.2.2.5 where parameters are constrained to fit along a specified function. We consider **Linear** $\{g_1(r(\alpha), r(\beta)) = r(\alpha) - r(\beta), g_2(r(\beta), r(\alpha)) = r(\beta) - r(\alpha)\}$ and **Exp**onential $\{g_1(r(\alpha), r(\beta)) = e^{-\kappa \cdot r(\alpha)} - e^{-\kappa \cdot r(\beta)}, g_2(r(\beta), r(\alpha)) = e^{-\kappa \cdot r(\beta)} - e^{-\kappa \cdot r(\alpha)}\}$ cases.

3.2.3.2 Citation Screening Results

For the citation screening datasets (see Table 2.2), we use a bag-of-words (BOW) representation, ignoring word capitalization and removing words found in the PubMed stop-list. During each experiment, we perform five-fold cross-validation, setting C_1 for each fold via two-fold cross-validation on the available training data for that fold (covering the search space $C_1 = 2^{\{-10,\ldots,3\}}$). Once C_1 is determined for the baseline SVM, we use the resulting $\mathbf{w}$ to inform $\tau_{\{+,-\}}$ and perform a grid (or cube) search over C_2, C_3, C_4, as appropriate.

[1] We set β to 10, reflecting intuition.

Furthermore, due to the class imbalance in the data, accuracy is not a useful metric for comparing models. Here we instead use a weighted harmonic mean which values recall more than precision, i.e., $F_2 = \frac{5 \cdot precision \cdot recall}{4 \cdot precision + recall}$. We have defined *recall* (also known as *sensitivity*)) and *precision* in Equations 2.1 and 2.3).[1] In reality we would increase the weight on recall in the citation screening scenario (although this would depend on the specific project), but we wanted to assess the performance of the CW-SVM with respect to less extreme asymmetry in costs. We note that in Chapter 4 we will introduce a metric specific to the citation screening case and a general method for eliciting the relative importance of sensitivity versus specificity from domain experts.

We use two of the systematic review datasets introduced in the preceding chapter (Table 2.2), namely proton beam (157) and COPD (32). The former comprises 4751 documents – 243 of which were screened in, i.e., labeled as *relevant*. Thus the minority class here comprises 243 documents. A clinician involved in the review provided seventy *positive* terms divided into six ranked term classes and eleven *negative* terms divided into three ranked term classes. The cost associated with term label acquisition is not reflected in the plots, in part because we believe would be quite low in this case as the reviewer knows these terms *a priori*.

We ran experiments as follows. We provided each learner access to the same datasets comprising an increasing number of the relevant citations. In particular, we included $\{50, 100, 150, 200, 243\}$ relevant citations in the five training datasets, respectively. For each of these training datasets, we ran five-fold cross-validation to assess classifier performances given the corresponding amount of training data.As per our discussions in Chapter 2, we undersample the majority instances such that we learn over a balanced dataset for all models.[2]. We report the average performances over this cross-validation for each amount of training data in Figure 3.3.

Not surprisingly (in our view) naïve Bayes fares poorly compared to the other

[1] We note that we use precision in place of specificity in this variant of F_2, in contrast to experiments in Chapter 2. The general trends using these two variants will be comparable: as specificity increases, so too precision.

[2] In order not to confound the analysis, we do not bag classifiers here.

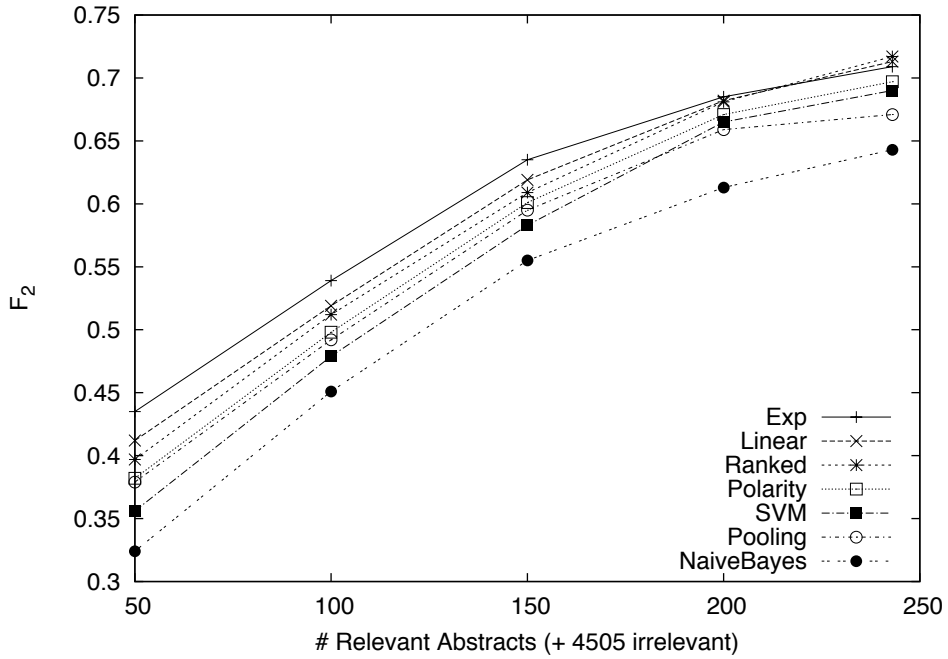


Figure 3.3: Empirical Results on the proton beam review

models. However, the pooling multinomials model (116) does relatively well, outperforming the standard SVM model, at least at the first three evaluation points. This demonstrates the utility of labeled features. All three of the CW-SVM models outperform the other strategies, particularly at the start of the learning curve (i.e., when fewer labeled instances are available). This makes sense, as biasing the learner with (prior) domain knowledge in the absence of sufficient training data seems likely to improve performance.

The second citation screening dataset that we use here is COPD, which comprises 1606 documents, 196 of which were found to be *relevant*. In this case, we have fifteen positive terms divided into three ranked term classes and seven negative terms divided into two ranked classes. For COPD, we conducted five experiments in which we learn a classifier from $\{40, 80, 120, 160, 196\}$ relevant examples and 1410 irrelevant documents. The experimental procedure is otherwise as described above. The results for this experiment are shown in Figure 3.4.

Naïve Bayes again performs poorly on the COPD dataset. Interestingly, the pooling multinomials does not perform well here, as it did above. It is not clear to us why this is the case, though it may be attributable to the comparatively small number of labeled features for this dataset. We again observe that the

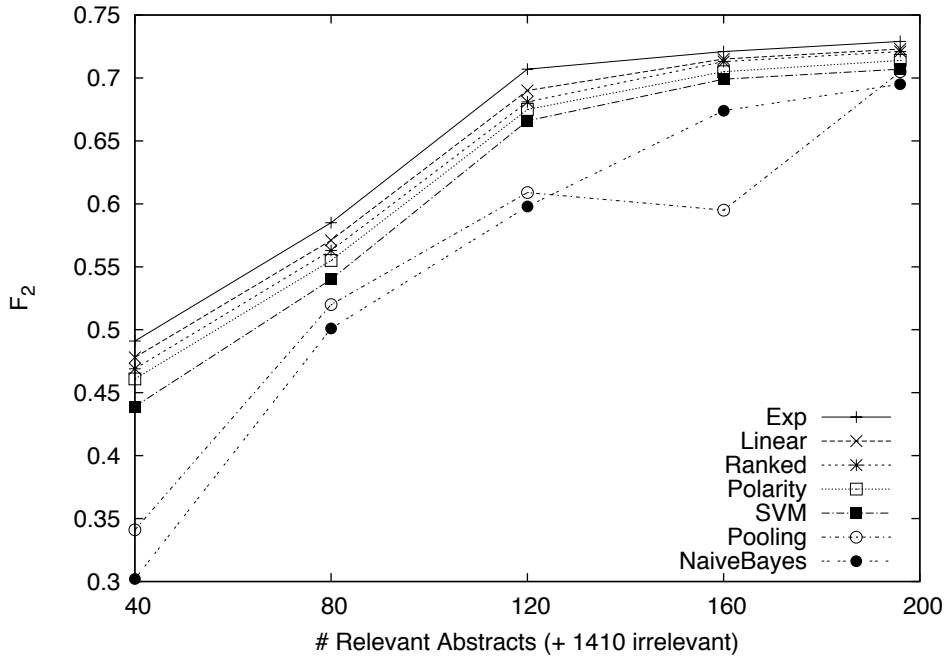


Figure 3.4: Empirical Results on the COPD review dataset.

CW-SVM outperforms all other models, particularly at the start of the learning curve – i.e., with fewer labels.

3.2.3.3 Sentiment Analysis Results

We now present results over the *movies* dataset (123), in which the task is to classify movie reviews as *positive* or *negative*. There are 2000 movie reviews in this corpus, 1000 of which are *positive* and 1000 of which are *negative*. For this dataset, we have rationales provided by Zaiden et al. (181), and we therefore compare against the annotator's rationales method described above. We follow the data encoding, training and testing procedures described in (181). To derive labeled features, we used an information-gain metric to rank terms with respect to their discriminative power using the instance labels to effectively simulate an oracle, as has been done elsewhere (57). Recall that the CW-SVM can exploit feature rankings. We thus created three classes of each polarity: thirty positive terms total (ten per positive rank) and 45 negative terms (fifteen per negative rank). We set $C_1, \ldots, C_4$ as described above.

Both standard naïve Bayes and linear pooling perform poorly in this case.[1]

[1]Our linear pooling results agree with those of (116), though our implementation of standard naïve Bayes outperforms theirs, for reasons obscure to us.

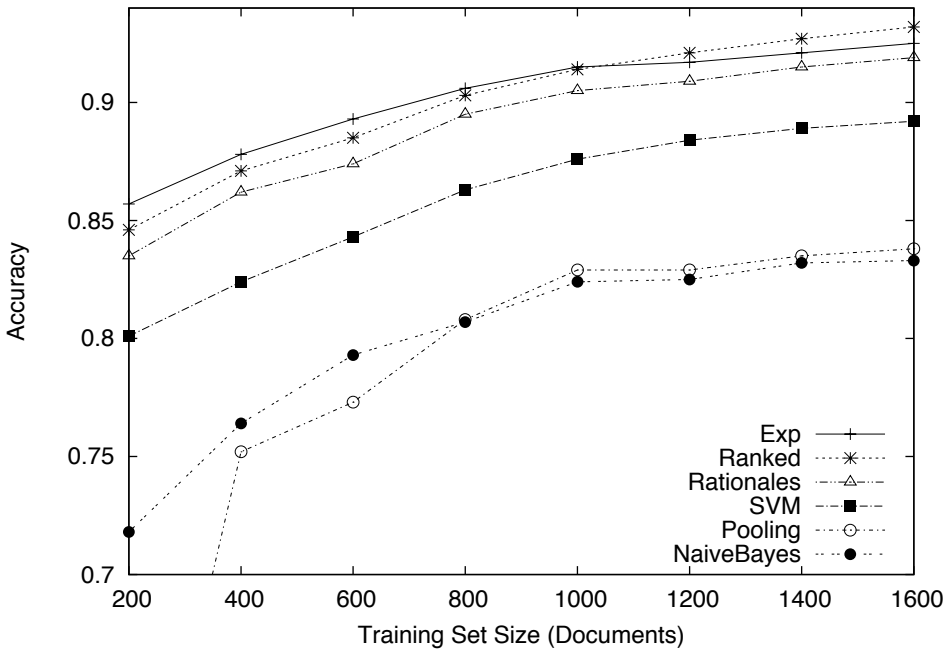


Figure 3.5: Empirical Results on Movies Dataset

All of the other strategies that exploit labeled features (our CW-SVM and the rationales approach) outperform the baseline SVM induced over instance labels alone, again highlighting the utility of labeled features. The CW-SVM, however, dominates the already strong rationales approach.

3.3 Conclusions

We have presented the CW-SVM, a novel, flexible method for directly incorporating labeled features in classifier induction. Our method requires only a small number of labeled features to outperform the baseline SVM. We presented strong empirical results, demonstrating that the CW-SVM outperforms existing methods that learn with labeled feature information over two biomedical abstract screening datasets and a sentiment analysis task.

Unlike existing dually supervised methods, which exploit only feature-class associations, the CW-SVM allows for the direct incorporation of *ranked* labeled features, allowing domain experts to impart knowledge regarding groupings of terms with varying degrees of polarity. As we saw in the experimental results, such rankings can boost classifier performance. Finally, should the expert have prior intuition regarding the relative polarities of sets of their labeled terms (e.g.,

weakly versus strongly positive), our framework provides a way of encoding this information.

We will later (Chapter 5) return to the paradigm of dual supervision in the context of *active learning*. But first we introduce active learning and address outstanding issues therein in the following chapter.

4

Real World Active Learning

In the preceding chapter, we focused on one mode of expert/model interaction that looks to improve model performance: labeled features. We now turn our attention to iterative, interactive learning. Specifically, in this chapter we exploit the *active learning* (AL) protocol (143) to induce better models with fewer labels, i.e., less human effort. In Chapter 5 we will present a method for jointly exploiting the dual supervision and active learning frameworks.

The trouble with existing active learning methods, which we will next review, is that they make a number of unrealistic assumptions. Specifically, they assume a single, infallible oracle will provide labels at a fixed cost. But in real-world scenarios, and indeed in citation screening for systematic reviews, it is often the case that multiple labelers participate in a given task, each with different abilities and costs. Moreover, it is not generally true that instances take an equal amount of time to label: difficult instances, for example, are likely to take more time – and hence cost more money – compared to instances that obviously belong to a given class. The main contributions of this chapter are novel active learning methods that address these problems, thus squeezing better performance out of fewer labels. Portions of this chapter appeared in the *Proceedings of the 2011 Siam Data Mining* conference (SDM 2011) (167) and in the *Proceedings of the 2010 International conference on Health Informatics* (IHI 2010) (163).

4.1 Background and Related Work

Active learning strategies exploit an expert 'in-the-loop' during classifier training. The aim is to make the induction process more efficient by allowing the learner to select its training data cleverly, rather than at random. Active learning has empirically proven quite successful in terms of expediting the training process (107, 145, 150). In this work we focus on *pool-based* active learning (100), in which the learner has access to a large unlabeled set (or *pool*) of instances $\mathcal{U}$ and a small labeled set of instances $\mathcal{L}$; the aim is to pick instructive examples from $\mathcal{U}$ for the expert to label. The pool-based scenario agrees with typical real-world cases: unlabeled data is often abundant and easily accessible (consider the Internet, for example), but tasking humans to annotate this data is costly. Active learning is particularly attractive in the citation screening case, because a new classifier must be induced for *each* project, as the target concept (i.e., the inclusion criteria) is different for each review.

4.1.1 Active Learning Methods

Aside from the pool-based framework, two other major variants of active learning have been studied at length, which we briefly review for completeness. In *stream-based* active learning (10, 47, 147), the learner is presented with instances sequentially and for each must decide whether or not to purchase its label. In contrast to pool-based active learning, it is usually assumed that the learner can consider each instance sequentially and only once. Aside from this crucial difference, however, the approaches used to decide which instances in $\mathcal{U}$ should be labeled in the pool-based scenario can also be used to decide for which instances a label should be acquired in the sequential or stream-based case.

Query-based learning strategies (5, 6, 7) allow the learner to create instances on its own and then ask the expert to which classes they would belong. The biggest drawback to such membership query learning strategies is the requirement that the learner generate coherent synthetic instances. Consider, for example, text classification. For a query-based strategy, the learner would need to generate documents that could reasonably be said to belong to a particular

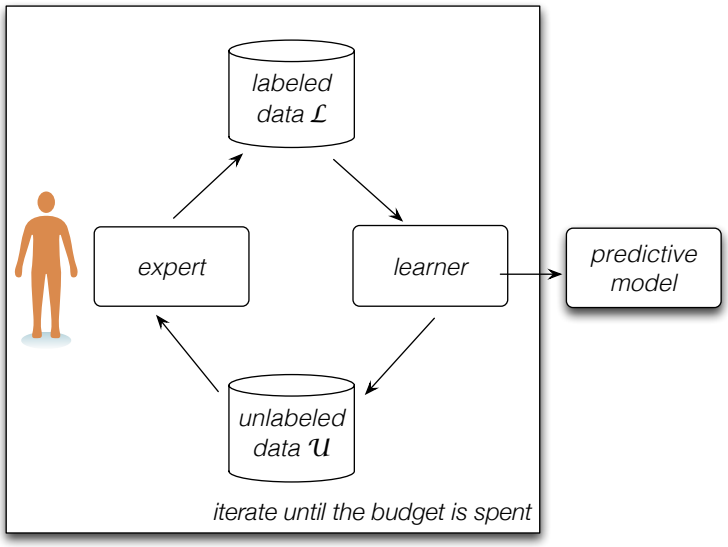


Figure 4.1: The pool-based active learning paradigm. The supervision in this case is iterative and interactive: at each step in the learning process, the model requests the expert to label instances whose annotation will likely lead to better predictive performance.

class. However, generating such coherent text is far beyond the capabilities of modern natural language processing (NLP) techniques. The requirement that humans classify synthetic instances is often problematic in practice, even outside of text classification (19).

Figure 4.1 describes the pool-based active learning process. In contrast to the standard supervised learning framework (Figure 1.3) in which the training data is selected at random up front, active learning is an iterative process. At each step in active learning, the model selects a small sample of instances from the unlabeled pool $\mathcal{U}$ for the human expert to label. The idea is that by selecting these instances intelligently, rather than at random, a better model can be induced with fewer labels. The instance selection strategy is a function of the current model. Generally, we can define a *query function* $\mathcal{Q}$ that selects from the remaining unlabeled instances $\mathcal{U}$ an instance $x*$ likely to be informative to the learner. Active learning methods differ with respect to their $\mathcal{Q}$ function. Pool-based AL methods (i.e., specifications of $\mathcal{Q}$) typically fall into one of three families of strategies: *uncertainty sampling*, *Query-by-Committee (QBC)* and *expectations-based* models.

Of these, uncertainty sampling (100) is the most widely used. The idea is sim-

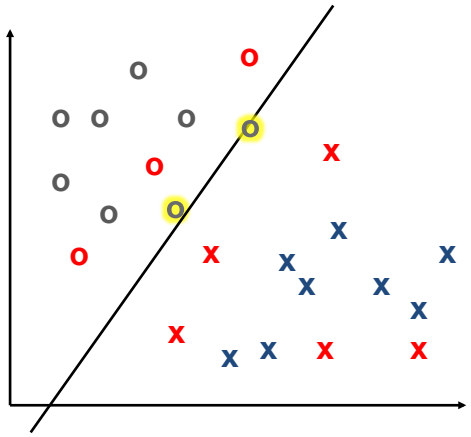


Figure 4.2: *Simple* Active learning with an SVM (158). Assume only red points are labeled, and that the remainder constitute $\mathcal{U}$. The drawn line shows what might be induced as the decision surface, given the (red) points labeled thus far. The two highlighted O's are nearest this surface, and are thus attractive candidates for labeling. This is intuitively agreeable here, as labeling these points will push the surface nearer the X's. One can see this will result in a better classifier for this data, because the shown hyperplane misclassifies one of two highlighted O's; labeling either of the highlighted candidates would remedy this.

ple: the model selects for labeling those instances about whose class membership it is least certain. Quantifying uncertainty is straight-forward for probabilistic models: in such cases the model is least certain about the point(s) with the lowest predicted likelihood of belonging to any of the classes $y \in Y$. Formally,

$$x* \leftarrow \underset{x \in \mathcal{U}}{\operatorname{argmin}} \max_{y \in Y} P(y|x) \tag{4.1}$$

Thus any model that explicitly estimates class membership probabilities – e.g., naïve Bayes, as in (90) – can immediately be used for uncertainty sampling.

In the case of discriminative models, uncertainty can be quantified via calibration methods (as were discussed in Chapter 2) or heuristically, contingent on the model. Perhaps the most popular flavor of uncertainty sampling is Tong and Koller's *Simple* (158), which uses Support Vector Machines (SVMs) (45) as the underlying learner. The intuition is to interpret the distance of an instance from the discriminating hyperplane as a proxy for confidence in its label (recall that these are the f_i values we used for calibration in Chapter 2). The querying strategy, then, is simply to pick the instance nearest the current hyperplane. Figure 4.2 illustrates this approach.

In recent work on uncertainty sampling methods, Dredze et al. (55) pro-

posed *confidence-weighted* active learning. This strategy leverages confidence-weighted linear classifiers (56) to attain a more fine-grained measure of uncertainty, compared to the aforementioned distance to the decision surface. In particular, confidence-weighted learners model each coefficient in a linear model as a normal distribution with an associated point estimate and variance. The uncertainty is then estimated via the *expected* distance to the decision boundary, which takes into account the variances around each w_j. This can be viewed as taking weighted draws from the space of 'good' hypotheses and aggregating their estimates of uncertainty with respect to each instance. The authors demonstrated that this approach outperformed standard uncertainty sampling strategies on some benchmark datasets.

Query-by-Committee (QBC) style algorithms constitute another family of active learning strategies (66, 147). These methods construct an ensemble of classifiers induced over $\mathcal{L}$ and request labels for instances in $\mathcal{U}$ about whose class said ensemble members most disagree. This strategy is theoretically motivated by computational learning theory (147): each committee member may be viewed as a hypothesis consistent with the instances comprising $\mathcal{L}$. Acquiring a label for an instance about which two or more hypotheses disagree can be seen, then, as a means of explicitly shrinking the version-space, i.e., the space of hypotheses consistent with $\mathcal{L}$: at least one of the models is incorrect, and can hence be removed from this space once the true label is revealed.

Many different variants of QBC have been proposed. For example, Mamitsuka experimented with boosting- and bagging-based methods (108). In boosting (65) one creates a set of classifiers iteratively, adjusting at each round the misclassification costs associated with instances in $\mathcal{L}$ on which errors were made in previous rounds. Bagging (26), meanwhile, draws some number of independent subsets (with replacement) from $\mathcal{L}$ and induces corresponding models over each of these samples. In both cases an ensemble of classifiers is produced, and can hence be used in the QBC framework. Another approach to QBC is to sample from the hypothesis space directly, as proposed for example by McCallum and Nigam (113). Elsewhere, Argamon-Engelson and Dagan (8) have extended the QBC paradigm to the class of probabilistic generative models. Specifically,

they measure uncertainty via the vote entropy calculated over their ensemble of models.

Expectations-based models form the last family of methods we will review here. Broadly, such methods look to explicitly maximize the change in some criteria of interest likely to be achieved by acquiring a label. For example, Roy and McCallum (137) propose selecting instances that directly minimize the expected resultant model prediction error. To this end, instances are given scores proportional to the reduction in error expected, should their label be revealed. This expectation is computed using the current model. Expected error minimization is theoretically appealing because it explicitly maximizes the quantity of interest, namely classification accuracy. It has also been shown to perform well empirically (137). Unfortunately, it is computationally intensive. In a similar vein, Cohn (43) proposed selecting the instance that results in the greatest (expected) reduction in variance. Finally, Settles and Craven (145) have suggested maximizing the *expected gradient length*. This approach looks to select the training instance(s) that will have the greatest effect on the parameters of the model being induced.

A major drawback to all variants of active learning is that they bias the training set by definition: no longer are the instances comprising the training set drawn i.i.d. from the underlying distribution. There have been a few recent efforts to mitigate this bias. Dasgupta and Hsu (49), for example, exploit hierarchical clustering to label sub-clusters of instances with bounded error-rates. Beygelzimer et al. (21), meanwhile, propose instance-weighted active learning (IWAL), in which the label for a given instance in $\mathcal{U}$ is requested with a probability carefully tuned to provide label complexity bounds.

Having reviewed the active learning framework and corresponding methods, we next address several unrealistic assumptions in active learning that have impeded its adoption for real-world tasks (14).

4.1.2 Unrealistic Assumptions in AL

In the canonical active learning scenario, one assumes that there is some budget available for acquiring labels from an expert. The expert is assumed to be *orac-*

ular, i.e., infallible. Acquiring a label from the expert has an associated cost. The aim is to spend the budget wisely in order to maximize predictive performance. The usual strategy is to greedily acquire labels for the most promising examples in $\mathcal{U}$ until the budget is exhausted. Looking to bring active learning out of theoretical development and into practice, in this chapter we introduce methods that relax the above assumptions. Our goal is to make AL more useful in real-world scenarios in general and in the citation screening task in particular.

We will start with the assumption that a single, infallible oracle provides labels requested by the learner at a fixed cost. Clearly this is unrealistic. Real-world applications suitable for active learning often include multiple domain experts who provide labels of varying cost and quality. Indeed, this is the case in the citation screening task: several reviewers typically participate in the screening for any given project, some of whom are usually seasoned systematic reviewers while others are relative novices. In Section 4.3 we explore this *multiple expert active learning* (MEAL) scenario and develop a novel algorithm for instance allocation that exploits the meta-cognitive abilities of novice (cheap) experts in order to make the best use of the experienced (expensive) ones. We demonstrate that this strategy outperforms strong baseline approaches to MEAL on both a sentiment analysis dataset and two datasets from our motivating application of citation screening. Furthermore, we provide evidence that novice labelers are often aware of which instances they are likely to mislabel.

A second unrealistic assumption we address in active learning is that instances are interchangeable with respect to labeling difficulty. That is, it is usually assumed that the cost (i.e., the required annotation time) of labeling a given example x_i is equal to that of labeling x_j $(i \neq j)$. But this is plainly naïve: usually certain instances will obviously belong to one class or another – and thus be easy to classify – while others will be more ambiguous and therefore require more time to label. The latter examples will be more expensive to label, though doing so may help to induce a better model compared to labeling easy instances. In Section 4.4 we develop a model to predict the time it will take to label instances and propose a means of incorporating these predictions into the active learning criteria. We show that incorporating these predictions into active

learning indeed improves performance: i.e., for the same fixed budget, one can induce a model with better predictive performance if (predicted) time-to-label is taken into account during active learning.

Recently, other researchers have begun to address these assumptions, too. Donmez and Carbonell (53), for example, have developed the *ProActive learning* framework for cost-sensitive active learning with multiple imperfect labelers. Theirs is a flexible decision-theoretic approach that looks to maximize expected utility. We compare this to our proposed *meta-cognitive* strategy for multiple expert scenarios in Section 4.3, and highlight issues in estimating the utilities associated with label acquisitions. Specifically, we show that the ProActive strategy does not account for workload distribution, and can suffer due to the difficulty in estimating the quantities required for its decision-theoretic calculations. We show that our approach outperforms ProActive strategies, at least in some cases.

Other recent research has also investigated empirical (real-world) annotation times. Arora et al. (9) demonstrated the feasibility of estimating the cost to label instances, even across different annotators, in a movie review classification task. As features, they incorporated information such as the word count (i.e., length) of a movie review. Their focus, however, was not on integrating this information into AL, but rather the annotation time prediction itself. Elsewhere, Baldridge and Palmer (15) emphasized the importance of taking annotator cost and expertise into consideration. They demonstrated that the efficacy of active learning can vary dramatically as a function of what measure of cost is used (e.g., number of labels provided versus the real annotation time), highlighting the need for time-sensitive active learning in real-world systems.

Settles et al. (146) demonstrated that knowing the (true) annotation time for unlabeled instances can theoretically improve active learning performance. However, the model they used to predict annotation times was not sufficient to improve performance, and thus when they used the *predicted* rather than the *true* labeling time, no performance gains were made. They used the same Return-on-Investment (ROI) strategy recently advocated by Haertel et al. (72), in which the utility calculated for an unlabeled example (i.e., a measure of its

informativeness due to the active learning scoring function $\mathcal{Q}$) is scaled by the the time it will likely take to label it. Haertel et al. demonstrated that factoring in predicted cost can improve active learning performance in a Part of Speech (POS) tagging task. They note that the difficult part is estimating the cost and utility functions. In Section 4.4, we present such functions for the citation screening task, and achieve strong empirical performance. But we first take a detour to consider the problem of appropriately evaluating the performance of active learning systems for real-world tasks, in order to later quantify the performance of the proposed methods.

4.2 Evaluating AL Systems in Imbalanced Scenarios

Evaluating classification systems in imbalanced scenarios, particularly in the context of active learning, is difficult to do correctly (60). Consider first the the aim of active learning, which is generally assumed to be deriving a good predictive model. But in many document retrieval applications (such as the citation screening case) the goal is to find all of the relevant instances in a *finite pool* (e.g., the set of citations retrieved with a PubMed search) rather than to induce a good predictive model. When presenting empirical results on systematic review datasets, we will be careful to appreciate the true aim: reducing the labor required to identify all of the relevant studies in a fixed set.

There is also the matter of using a suitable metric to quantify model performance. Most work in information retrieval (IR) concerning metrics for the evaluation of text classifiers has focused on variants of the weighted F-measure (99), i.e., the weighted harmonic mean of sensitivity and precision (we will sometimes use specificity rather than precision).[1] This weighting is parameterized by λ, which explicitly encodes the tradeoffs inherent in the scenario under consideration. Following this tradition, we will assume that $cost(fn) = \lambda \cdot cost(fp)$ for some λ.

In the preceding chapter, we somewhat arbitrarily set $\lambda = 2$, effectively weighing sensitivity four times as much as precision. This was an attempt to

[1] Taken together, sensitivity and specificity provide more information than sensitivity and precision because specificity is independent of sensitivity, whereas precision is not.

evaluate learning strategies for imbalanced scenarios in a general way using a common metric. The correct λ for a given task is inherently application-specific. But asking domain experts to provide this parameter outright is probably not a good idea; humans, in general, are poor at this sort of explicit quantification (86). Rather, we would like to *elicit* this weighting in a natural way. To this end, we propose appropriating a method from medical decision theory (160) for eliciting this weight from domain experts that accounts for imbalanced classes and asymmetric costs. We then apply this method to the case of citation screening to define an appropriate λ.

Suppose that a predictive model – or an oracle – provides the probability that a given citation is irrelevant (more generally, that a specific instance belongs to the majority class). If this probability is sufficiently low, a rational reviewer will want to peruse the abstract in full to ascertain if it should be included. On the other hand, if the probability is high enough, a reviewer will not bother to read the abstract. There is some threshold probability p_t at which the reviewer forgoes reading the abstract. In other words, reviewers are at this point indifferent to whether or not they read the abstract because, at this threshold, the expected value of their reading it is equal to the expected value of their not reading it. Suppose that we elicit this p_t from the expert. Further, let $\mathcal{V}(tp)$, $\mathcal{V}(fp)$, $\mathcal{V}(fn)$, and $\mathcal{V}(tn)$ denote the value of a true positive, false positive, false negative and true negative, respectively. We have:

$$p_t \cdot \mathcal{V}(tp) + (1 - p_t) \cdot \mathcal{V}(fp) = p_t \cdot \mathcal{V}(fn) + (1 - p_t) \cdot \mathcal{V}(tn) \tag{4.2}$$

The LHS of Equation 4.2 is the expected value of reading the abstract; the RHS is the expected value of not reading the abstract. This implies:

$$\frac{\mathcal{V}(tp) - \mathcal{V}(fn)}{\mathcal{V}(tn) - \mathcal{V}(fp)} = \frac{1 - p_t}{p_t} = \lambda \tag{4.3}$$

Then $\mathcal{V}(tp) - \mathcal{V}(fn)$ is the penalty of not reading a relevant abstract, and $\mathcal{V}(tn) - \mathcal{V}(fp)$ is the cost associated with reading an irrelevant abstract. Thus $\frac{1-p_t}{p_t}$ is the ratio of the cost of a false negative to the cost of a false positive, giving us

our desired λ. Recall the definitions of *sensitivity* and *specificity* from Equations 2.1 and 2.2. We then define our metric, which we call $Utility_\lambda$, as follows:

$$\frac{\lambda \cdot sensitivity + (1 - specificity)}{\lambda + 1} \tag{4.4}$$

For evaluation of models for the task of citation screening, we elicited this weighting from the project lead on one of the ongoing systematic reviews at the Tufts EPC. We asked him at what probability of a document being irrelevant would he exclude it without reading the abstract. We asked this same question repeatedly, increasing the number of citations that needed to be screened for the hypothetical project. Note that this is a more intuitive question to answer than that of an explicit request for λ, because it mimics a real-life decision process. In line with expectations, p_t decreased slightly as the set of citations that needed to be screened grew. Specifically, for $N \leq 10,000$ abstracts, the threshold p_t provided was 95% of being irrelevant ($p_t = .05$), which translates to $\lambda = 19$. When $N > 10,000$, he set $p_t = .1$ (90% of being irrelevant), giving $\lambda = 9$. In general, we will use λ=19 in our experimental evaluations, because the systematic review datasets with which we experiment (Table 2.2) comprise 10,000 or fewer citations.

Now that we have defined a suitable evaluation metric, we turn our attention back to methods for real-world active learning. We will make use of the U_{19} metric in our empirical evaluations of the proposed approaches.

4.3 Multiple Expert Active Learning

A significant obstacle to deploying supervised machine learning systems is obtaining sufficient labeled training data to achieve acceptable performance. The active learning protocol looks to mitigate this expense by allowing the learning algorithm to interactively choose its training data from an unlabeled pool with the aim of selecting only those examples most useful in inducing a model. But, as discussed in 4.1.2, active learning methods have tended to make several unrealistic assumptions. One such simplifying assumption is that labels are provided

by a single, infallible oracle. This is clearly not always the case – often a group of annotators can provide labels of varying quality and cost.

In this section, we investigate this *multiple expert active learning* (MEAL) scenario, wherein a group of domain experts, each with an associated cost and level of expertise, participate in the active learning task. We explore a fundamental problem in this common real-world scenario that has thus far received limited attention: given a panel of experts, a set of unlabeled examples and a budget, who should label which examples?

Ostensibly, the MEAL scenario is similiar to 'crowd-sourcing' (e.g., Amazon Mechanical Turk[1]), in which annotation tasks are performed at some cost by a (typically anonymous) group of users via a task marketplace. Recent work has begun to explore active learning strategies for this scenario in the context of machine translation (4). Our case differs in an important way – we are interested in settings in which all annotators must possess a requisite minimum aptitude for annotating instances, precluding the use of low-cost, untrained annotators via crowd-sourcing. This setting corresponds to a relatively common scenario, particularly in 'specialized' (e.g., scientific/biomedical/linguistics) domains: multiple domain experts with varying levels of expertise/experience and commensurate costs are to annotate a pool of data. In such scenarios, the objective is to derive an active learning querying strategy that assigns instances appropriately with respect to annotator expertise and expense, i.e., we would like to assign 'easy' instances to novice experts and 'difficult' instances to skilled experts.

The remainder of this section is structured as follows. First, we introduce and motivate the MEAL scenario in the context of domains in which annotation workload must be balanced across multiple experts of varying aptitude and cost. In Section 4.3.1, we identify deficiencies of related MEAL approaches, and then develop a novel algorithm for MEAL in Section 4.3.2 that exploits the meta-cognitive abilities of novice labelers to inform the allocation procedure. Intuitively, our approach relies on inexpensive experts to flag difficult examples encountered during annotation; these will subsequently be reviewed by more ex-

[1] http://www.mturk.com

perienced experts. We demonstrate empirically that this strategy out-performs strong baselines, including previously proposed strategies for MEAL (53), with respect to a sentiment analysis task (123) (4.3.3) and our motivating scenario of biomedical citation screening. Further supporting the proposed approach, in Section 4.3.4 we provide empirical evidence that novice labelers are indeed conscious of which examples they are likely to mislabel, and also argue that automatically identifying difficult instances is a hard task.

4.3.1 Proactive Learning and Baseline Strategies

We begin by reviewing ProActive learning (PAL) (53), a previously proposed framework for practical active learning (including multiple expert scenarios). One of PAL's strength is its flexibility; it selects expert/instance pairs at each step in active learning decision-theoretically, so as to maximize expected utility. We find quantifying this utility tricky. Specifically, in practice it is difficult to reliably estimate the variables required for estimating utility, as we discuss below. In 4.3.1.2 we introduce two new, simple baseline strategies for MEAL: random and active random.

4.3.1.1 ProActive Learning

ProActive learning (PAL) was proposed by Donmez and Carbonell (53, 54) as a decision-theoretic approach to the task of selecting expert-example pairs during each round of active learning. Denoting the instance space by $\mathcal{X}$ and (fixed) set of experts by $\mathcal{E}$, PAL requires the specification of a value of information function, $\mathcal{V} : \mathcal{X} \rightarrow \mathbb{R}$, that maps instances to their expected utility with respect to inducing a classifier, and an expert-specific cost value $C : \mathcal{E} \rightarrow \mathbb{R}$.

Typically in active learning, one defines $\mathcal{Q}$, which maps a given unlabeled instance to a scalar representing the expected value of acquiring its label. For example, uncertainty sampling scores unlabeled instances based upon how uncertain the model is in their predicted labels. Donmez and Carbonell propose that $\mathcal{V}$ can be any such active learning scoring function. Specifically, again de-

noting the pool of unlabeled instances by $\mathcal{U}$, for each iteration of PAL, an expert and example is selected according to:

$$(e^*, x^*) = \underset{e \in \mathcal{E}, x \in \mathcal{U}}{\operatorname{argmax}} \frac{p(e, x) \cdot \mathcal{V}(x)}{C_e} \tag{4.5}$$

When empirically comparing our proposed methods to PAL, we use two variants of Equation 4.5 (both of which were proposed by Donmez and Carbonell (53)), that imbue $p(e, x)$ with different semantics. For the sentiment analysis task, we set $p(e, x)$ to the probability of expert e providing a correct label for example x (i.e., Algorithm 2 in (53)). Because it is unclear how to estimate this probability in our experiments, we 'cheat' in favor of PAL by using the true probability from the generative model used to derive the experts. The second variant defines $p(e, x)$ as the probability that expert e will provide a label for instance x (i.e., Algorithm 1 in (53)). We use this version for our citation screening experiments, where we actually have multiple real-world experts. We note that if one is interested in quantifying the expected utility as defined as improvement over labeling instances as belonging to the predominant class, then one should adjust $p(e, x)$ for prevalence, i.e., by using $p(e, x) - \pi$. Here we are interested in using these utility scores to rank instances, and thus adjusting by a constant such as π will have no effect.

In this case, examples that the novice reviewer labeled as 'difficult' were treated as instances this expert refused to label within the PAL framework. We then induced a classifier to predict the probability of the novice expert providing a label for a given instance.[1] As we will observe, one shortcoming of PAL for our setting is that the expert-example pair is selected myopically, without regard to balancing workload at each step, often resulting in inequitable workloads for participating labelers.

4.3.1.2 Random Baselines

The other two strategies we compare to are both randomized variants that assign examples to experts selected with equal probability. The simplest instance of this

[1] We used a linear kernel SVM, estimating $p(e, x)$ by scaling the output via Platt's method (127).

is the most straight-forward MEAL strategy possible: pick both the example and the expert at random. We refer to this method simply as *random*. Note that this is just (passive) learning with labels provided by multiple sources wherein the model ignores which annotators labeled what. The second strategy, which we refer to as *active random*, selects instances in decreasing order of $\mathcal{Q}(x_i)$ (i.e., estimated informativeness) and again picks experts uniformly at random.

4.3.2 Meta-Cognitive MEAL

We now present our strategy for MEAL, which comprises two technical innovations. First, unlike PAL, we explicitly model the workload distribution to ensure that all available annotators are assigned a sufficient amount of work – this is important in our scenario wherein experts are also possibly paid when not labeling data (i.e., salaried). Second, we allocate instances commensurate with expertise in a cost-effective manner by exploiting the meta-cognitive abilities of novice labelers. In particular, we augment the binary label set $\{-1, 1\}$ with a third, extra-categorical label of "difficult", i.e., the annotator is unsure how to categorize the instance because it is too hard. When an expert labels an instance as "difficult", it is passed on to someone with more expertise.[1] Note that we assume this extra-categorical label incurs the same annotation cost as providing any other label. Asking highly skilled experts to *re-label* difficult instances makes sense in light of recent work by Sheng et. al (149) in which they argued that it is often more worthwhile to re-label instances rather than to label as-yet unlabeled examples. Rebbapragada has also shown that sometimes asking experts to spend more time thinking about particular, seemingly noisy labels they previously provided can be more fruitful than acquiring new labels (133).

This exploitation of human intelligence during active learning bears some resemblance to Attenberg and Provost's recently introduced *guided active learning* (13), wherein the expert explicitly provides instances from the minority class in active learning scenarios with class imbalance. In meta-cognitive MEAL we similarly rely on human intelligence, specifically by assuming that experts are

[1] We allow all except for the most highly skilled expert to use the "difficult" label; were s/he to label an example as such, there would be no one more qualified to whom we could defer.

capable of identifying difficult instances selected for labeling by the learner. (By contrast, in guided active learning it is assumed that experts are able to find representative instances of the minority class.) More precisely, we assume that novice experts will refuse to provide a label when they have low confidence in their ability to correctly classify a given example. In this case, we defer to someone with more expertise to label said instance. We provide empirical evidence that inexperienced labelers are indeed aware of which instances they are likely to mislabel in Section 4.3.4.

To allocate instances, we require an estimate of each expert's level of expertise throughout the MEAL process, denoted by α_e. In practice, these may be inferred via unsupervised methods (e.g., using EM) over a small sub-sample of instances labeled by all participating experts (54, 132, 172), or through available domain information such as expert salaries. We take the latter approach in this work, as we assume that within effective organizations, expert pay grade is highly correlated with aptitude. We note that this assumption will not always be valid, e.g., in cases where pay may correlate only with seniority. In such cases expertise should be estimated either via the aforementioned unsupervised methods, or perhaps via other external domain information (e.g., knowledge regarding individual expertise levels).

Algorithm 1 Meta-cognitive MEAL: Allocation, Version A

Input: Unlabeled data $\mathcal{U}$, active learning scoring function Q, expert panel $\mathcal{E}$, per-round labeling budget $\mathcal{B}$, desired work distribution parameters $\mathcal{W}$, expert queues q_e $\forall e \in \mathcal{E}$

$\mathcal{U} \leftarrow$ sort $\mathcal{U}$ according to Q
$\mathcal{C} \leftarrow 0$ {total annotation cost}
$\mathcal{A} \leftarrow \{\}$ {list of assignments}
while $\mathcal{C} < \mathcal{B}$ **do**
  $e^* \leftarrow$ draw from $\mathcal{E}$ according to $Mult(\mathcal{W})$
  if $|q_{e^*}| > 0$ **then**
    $x^* \leftarrow q_{e^*}.dequeue()$
  else
    $x^* \leftarrow$ next example in $\mathcal{U}$
  end if
  $\mathcal{A} \leftarrow \mathcal{A} \cup Assign(e^*, x^*)$
  $\mathcal{C} \leftarrow \mathcal{C} + C_{e^*}(x^*)$
end while
Output: List of assignments $\mathcal{A}$

Algorithm 2 Meta-cognitive MEAL: Re-Allocation

1: **Input:** Expert panel $\mathcal{E}$, expert queues q_e $\forall e \in \mathcal{E}$, example x^*, expert to whom example $x*$ was originally assigned to e^*, estimates of expertise levels α_e $\forall e \in \mathcal{E}$

2: $\mathcal{E}' \leftarrow \{e | e \in \mathcal{E}, \alpha_e > \alpha_{e^*}\}$
3: draw e' from $\mathcal{E}'$ with $p \propto \alpha_{e'}$
4: $q_{e'}.enqueue(x^*)$

Our strategy is presented in Algorithms 1 and 2. Note that Algorithm 2, Re-Allocation, is invoked whenever a novice labeler designates an instance as difficult. Note also that assignments are different from expert queues. The latter holds instances that less-experienced experts decided not to label (due to difficulty); this will always be empty for the most-novice expert. The key insight is to rely on the ability of the novice expert to identify the challenging examples that the strong expert ought to label. The benefits of such a strategy are two-fold; weak experts will label easier examples at a low-cost while expensive experts will be used sparingly and wisely on difficult examples. To achieve this, we first sort the unlabeled pool of documents, $\mathcal{U}$ by an active learning scoring function $\mathcal{Q}$ (Line 2). At each MEAL step, we draw from a multinomial parameterized by $\mathcal{W}$ to select an expert (Line 6). This distribution may either reflect a preference for equitable labor shares or may be dynamically updated to maximize some other objective.[1] For example, in our sentiment analysis experiments, we initially set $\mathcal{W}$ such that $w_e \propto C_e$. As soon as a weak expert has refused to label a difficult example (i.e., when there exists a non-empty q_e), we set $w_e \propto |q_e|$, that is, proportional to the size of the (stronger) expert's queue of re-assigned (difficult) instances, thus prioritizing the re-labeling of hard instances over the labeling of unlabeled examples. Once the queues are exhausted, we return to distributing examples to experts with probability inverse to their cost. If the drawn expert's queue of re-assigned examples is not empty, then they are assigned the next instance from their queue. Otherwise, they are assigned the next instance in the ranked pool, $\mathcal{U}$ (Lines 7-10). We continue until the per-round budget is exhausted. Algorithm 2 describes our re-allocation strategy. When a relatively

[1] If updated, however, one must be careful to renormalize so that $\mathcal{W}$ sums to 1.

novice expert designates an example as being difficult, it is assigned to a more experienced expert's queue with probability proportional to their expertise.

Algorithm 3 Meta-cognitive MEAL: Allocation, Version B

1: **Input:** Unlabeled data $\mathcal{U}$, labeled data $\mathcal{L}$, active learning scoring function Q, probabilistic classifier induced over 'trusted' labeled examples g, expert panel $\mathcal{E}$, per-round labeling budget $\mathcal{B}$, desired work distribution $\mathcal{W}$, expert queues $q_e \; \forall e \in \mathcal{E}$

2: $\mathcal{U} \leftarrow$ sort $\mathcal{U}$ according to $\mathcal{Q}$
3: $\mathcal{C} \leftarrow 0$
4: $\mathcal{A} \leftarrow \{\}$ {list of assignments}
5: **while** $\mathcal{C} < \mathcal{B}$ **do**
6: $\quad e^* \leftarrow$ draw from $\mathcal{E}$ according to $Mult(\mathcal{W})$
7: $\quad$ **if** $|q_e^*| > 0$ **then**
8: $\quad\quad x^* \leftarrow q_e^*.dequeue()$
9: $\quad$ **else**
10: $\quad\quad c \leftarrow$ flip a coin with bias $\propto \alpha_e$
11: $\quad\quad$ **if** c is heads **then**
12: $\quad\quad\quad x^* \leftarrow \arg\min_{x \in \mathcal{L}} g(label(x)|x)$
13: $\quad\quad$ **else**
14: $\quad\quad\quad x^* \leftarrow$ next example in $\mathcal{U}$
15: $\quad\quad$ **end if**
16: $\quad$ **end if**
17: $\quad \mathcal{A} \leftarrow \mathcal{A} \cup Assign(e^*, x^*)$
18: $\quad \mathcal{C} \leftarrow \mathcal{C} + Cost(e^*, x^*)$
19: **end while**

20: **Output:** List of assignments $\mathcal{A}$

In Algorithm 1, if no instances designated as difficult by lesser experts are assigned to the drawn expert, we then assign to them the next instance in the ranked unlabeled pool, $\mathcal{U}$ (line 8). However, depending on the scenario, it may make more sense to re-label instances in the labeled pool $\mathcal{L}$ with some probability, even though these examples were labeled with relatively high confidence by definition. This makes sense in cases where incorrectly annotated training data is costly, even if it doesn't affect the predictive performance of the induced model. Algorithm 3 operationalizes this intuition.

The key difference between this and Algorithm 1 is the procedure for when the drawn expert's queue is empty (Lines 9-15). Formerly, they were simply assigned the next instance in the sorted $\mathcal{U}$. Here they are assigned a labeled instance in $\mathcal{L}$ with probability proportional to their estimated accuracy α_e (Lines 10-12). In particular, we maintain a probabilistic model g induced over the highest skilled expert. We then select for labeling the instance in $\mathcal{L}$ whose assigned label has

the lowest probability of being the true label, according to g (Line 12). This can be viewed as attempting to automatically identify mislabeled instances in $\mathcal{L}$ in a semi-supervised way in multiple expert scenarios.[1]

4.3.3 Empirical Results – Simulation Experiments with Sentiment Analysis Data

We first present an experimental evaluation over a sentiment analysis task (used in the previous chapter). This benchmark dataset (123) has only a single set of gold-standard labels. To compare MEAL strategies, we therefore must generate artificial experts to simulate multiple labelings. The aim of this experimental setup is to demonstrate that when novice reviewers are capable of recognizing those instances they are likely to mislabel, the meta-cognitive MEAL strategy (Algorithm 1) outperforms strong baselines in terms of induced model performance versus cost. We justify these assumptions empirically in Section 4.3.4, in which we show that novices are indeed capable of discerning difficult examples. Furthermore, in Section 4.3.8 we show that our strategy outperforms baseline strategies in practice (i.e., in the citation screening task).

Recall that the movie sentiment dataset was created by Pang and Lee (123). Further recall that this dataset comprises 2000 movie reviews, half of which have been designated as 'positive' and the other half as 'negative'. The aim is to induce a classifier to discriminate between positive and negative reviews. This movie sentiment data is attractive for our work because it is a widely utilized classification task. Moreover, due to the subjectivity inherent to the task, one can easily envision variance in expert ability to categorize reviews.

To model the MEAL scenario, we must simulate labeling of the dataset by multiple experts with varying cost and skill. Moreover, we must associate a measure of difficulty with each instance. To this end, we use the probabilistic model for multiple annotators proposed by Whitehill et al. (172). In particular, we assume that each expert e has an associated expertise level $\alpha_e \in (-\infty, \infty)$, where large α_e implies a skilled labeler. Furthermore, we assume that each

[1] Note that this has connections to work on automatically identifying mislabeled instances (29); the difference here is that we are explicitly modeling one (experienced) expert to infer the mistakes of another (inexperienced) one.

instance x has an associated difficulty $\beta_x \in [0, \infty)$ where small β_x implies a difficult example. Following Whitehill's notation, we denote the label given by expert e to example x by $\hat{y}_{ex}$ and the true label for example x by y_x. Then probability with which expert e labels instance x correctly is as follows:

$$p(\hat{y}_{ex} = y_x | \alpha_e, \beta_x) = \frac{1}{1 + e^{-\alpha_e \beta_x}} \tag{4.6}$$

For our experiments, we generated both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. To set $\boldsymbol{\beta}$, we begin with the observation that in the citation screening task, instances can be categorized roughly into two categories; hard and easy examples. We believe this to be a more general phenomenon (and hence applicable to the movies dataset, for example), as is consistent with observations made by Beigman et al. (20). We thus invent two Gaussian distributions over $\boldsymbol{\beta}$; one corresponding to hard and the other to easy examples. We believe that the majority of the easy instances will in fact be extremely easy, as in our experience the majority of examples in classification problems fall obviously into a specific class. To model this, we truncate the Gaussian corresponding to the easy examples at its mean (see Figure 4.3), thus most examples will be relatively quite 'easy'. (If we did not truncate, the mean would be shifted away from the 'easiest' examples and toward 'medium-difficulty' instances).

We arbitrarily decided that $p_{easy} = .6$ of the instances were to belong to the easy class. Thus a β_x for each example x was drawn from the easy distribution with probability p_{easy} and from the hard distribution with probability 1-p_{easy}. Figure 4.3 shows a histogram of $\boldsymbol{\beta}$ drawn for the 2000 movie review instances. The expertise levels, $\boldsymbol{\alpha}$, were generated under a similar assumption. As we are

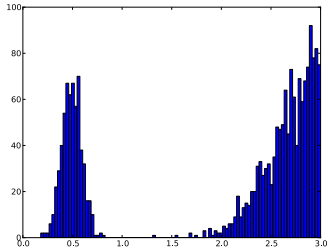


Figure 4.3: Histogram of drawn βs.

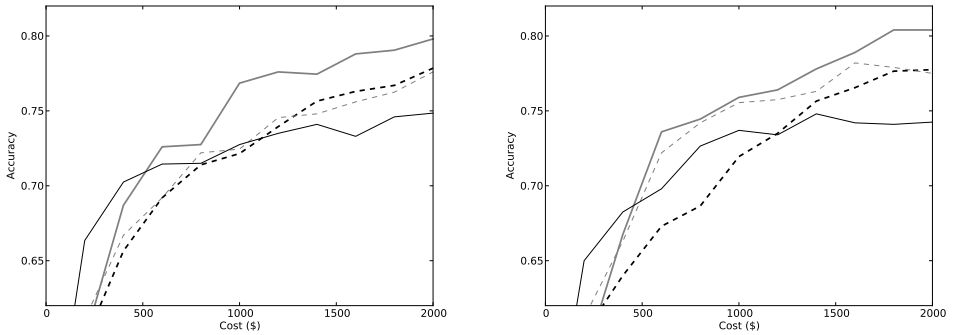
interested in scenarios wherein the participating annotators have varying levels

of expertise and cost, we generate experts belonging to two classes; weak and strong. These correspond to novice and experienced labelers, respectively. In reality, there may be more of a gradient in expertise levels, but this bimodal distribution captures the essence of the situation in which we are interested. Furthermore, for specialized domains (where active learning is arguably most valuable, because if labeling does not require domain expertise, labels can likely be acquired cheaply) such a binomial distribution is reasonable, because it encodes the trainer/trainee relationship common in such work. The $\boldsymbol{\alpha}$ values are thus drawn from Gaussians with means set such that the average probability of correctly labeling a given difficult example under the above proposed model is 0.6 and 0.95, respectively. We set the corresponding variances to 0.1. Likewise, we draw a salary for each weak and strong expert from two Gaussians, with means \$30,000 and \$150,000, respectively, both with variances of \$10,000. Note that we don't require salary to be a perfect predictor of labeler accuracy, but rather a crude proxy. We assume the weak experts designate an instance as 'difficult' when the (true) probability of their labeling it correctly is $\leq .8$.

Figure 4.4 shows results with a varying number of (simulated) participating experts. The y-axis in all plots corresponds to the induced accuracy over a hold-out set,[1] and the x-axis to cost. We compute cost by multiplying the expected time to label an instance (movie review) by the unit cost of the labeler, as calculated from their salary. To calculate a reasonable time to label for each review, we make the simplifying assumption that all labelers read 250 words per minute and transform the length of a review, as measured by word-count, to a labeling time under this model. All plots shown are averages over ten-fold cross-validation.

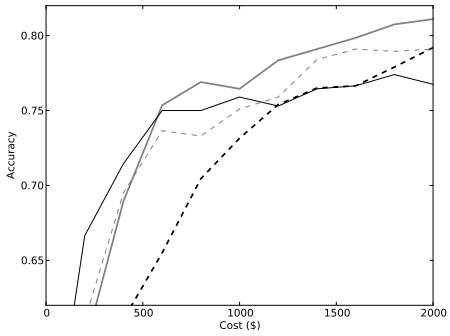
The main observation to make is that after the \$500 mark, the meta-cognitive curve dominates all other strategies, in all four simulated scenarios. The difference in induced accuracy is particularly pronounced in the two-expert case. It is also interesting that the active random strategy tends to outperform the ProActive strategy. We believe this is due to the greedy nature of the latter, which

[1]Note that accuracy is the correct metric to use in this case, because the class distribution is balanced.

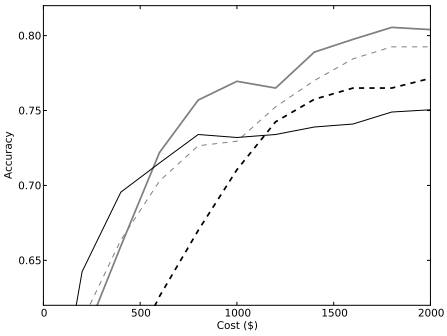


(a) Two experts (1:1)

(b) Three experts (2:1)



(c) Five experts (3:2)



(d) Ten experts (7:3)

Figure 4.4: Results over movies dataset with synthetic experts. The number of 'weak':'strong' experts, respectively, is given in the parentheses beneath each plot. The four strategies shown in each plot are: meta-cognitive MEAL (the solid, thick grey line); ProActive learning (53) (the bold, dotted black line); active random (the dotted grey line); random (the thin, solid line). One interesting phenomenon seen in these plots is that for low dollar amounts (< 500), random sampling consistently outperforms other methods. It is not entirely clear to us why this is the case, but one explanation may be that random sampling effectively acquires a relatively cheap set of labels from a diverse set of experts with little money, while other strategies spend their allotments relatively quickly. There may be an advantage very early on in acquiring many labels cheaply; but notice that this strategy quickly asymptotes.

we have observed to query the most expensive expert(s) nearly exclusively, thus acquiring far fewer instance labels (see Figure 4.5).

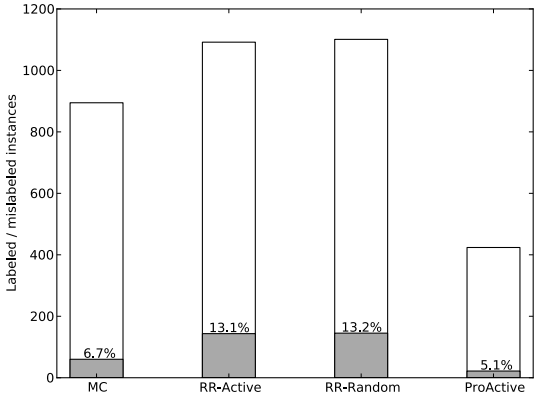


Figure 4.5: Number of unique instances that were labeled correctly (white) and the number that were mislabeled (grey), for each strategy.

Because imperfect labels are cheap, there is a trade-off between acquiring as many labels as possible and introducing mislabeled instances into the training set. This is shown in Figure 4.5, which plots the average number of unique instances that were labeled, and the percentage of those that were mislabeled once the budget was exhausted. Notice in particular that, unsurprisingly, the two random baseline strategies acquire the most unique instance labels, though they also incur the highest percentage of mislabeled instances in their training sets, at $\sim 13\%$.

4.3.4 On The Dunning-Kruger Effect

We have shown that our meta-cognitive strategy for MEAL can outperform other approaches if the (novice) labelers are capable of identifying the instances that they are likely to mislabel. This assumption of self-awareness regarding annotation acumen seems at odds with the known tendency for lower-skilled individuals to over-estimate their abilities, a phenomenon known as the Dunning-Kruger effect (92).

In the seminal paper on the subject, Dunning and Kruger provide evidence for the following conjecture:

> ... the skills that engender competence in a particular domain are often the very same skills necessary to evaluate competence in that domain ... (92)

If this were indeed the case it would be problematic for the proposed meta-cognitive approach; surely if novice reviewers are unable to recognize instances they are likely to mislabel, then the strategy cannot be effective. A natural concern is that due to the Dunning-Kruger effect, only skilled experts will be able to recognize 'difficult' instances. However, in the following section we provide preliminary evidence that at least for our biomedical citation screening domain, novice labelers are indeed capable of recognizing those instances that they are likely to mislabel (i.e., difficult instances).

4.3.4.1 Labeling Confidence

We first explore whether the confidence annotators have in their provided labels correlates with the likelihood of the labels being correct. To this end, we asked two novice reviewers to provide the 'confidence' they place in their own labels, and compared the average confidence ratings between the examples they classified correctly and the examples they misclassified.

Specifically, the two novice reviewers (we refer to them as Reviewers "1" and "2") screened 4751 biomedical abstracts from the proton beam dataset (157) summarized in Table 2.2. Note that we didn't use the proton beam dataset in evaluation because we did not prospectively gather 'difficult' labels from novice reviewers at the time of their screening the citations. Using the labels of a third, senior expert as a 'gold standard' we identified for each novice reviewer their sets of true positive, true negative, false positive, and false negative abstracts (denoted TP, TN, FP, and FN, respectively). For each reviewer we selected a manageably-sized random sample of citations stratified over these four sets. Because the sizes of the four sets are very different (there are many more false positives than false negatives), we used the following weighted random sampling scheme: we selected all $|FN|$ examples in the FN set (the smallest set); twice as many examples ($2|FN|$) from the next more prevalent set (FP); and thrice as many examples ($3|FN|$) from each of the remaining most prevalent sets (TP, TN). The stratified random samples consisted of 198 citations for Reviewer 1, and 171 for Reviewer 2.

We then presented the novice reviewers with the sampled citations in random

order, along with the labels they had provided, and asked them to rate for each citation their confidence in their own labels using a four-point scale of equispaced categories, i.e., a Likert scale (102): 'very uncertain', 'uncertain', 'certain', and 'very certain'. For analysis, we encoded the four categories as $\{-2, -1, 1, 2\}$, respectively. The reviewers were blinded to the 'gold standard', and were told that they were given a random sample of citations with no details regarding the sampling scheme. We tested whether the mean confidence of each reviewer differed between the citations they classified correctly (TP and TN) or incorrectly (FP and FN) using linear regressions accounting for the probability sampling weights of our sampling scheme.

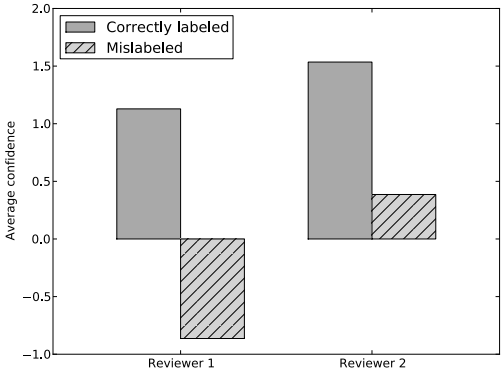


Figure 4.6: Average (novice) annotator confidence provided for labels of both correctly and incorrectly labeled examples over the proton beam dataset.

Figure 4.6 shows the mean confidence scores for correctly and incorrectly classified examples extrapolated to the total corpus for both reviewers. Mean confidence scores for Reviewer 1 were 2.6 units (95% confidence interval: 2.2, 3.0) higher in the correctly classified citations compared with the incorrectly classified ones; for Reviewer 2, the difference was 1.1 units (95% confidence interval: 0.6, 1.5). The difference is statistically significant for both reviewers ($p < 0.0001$). In other words, novice reviewers were substantially more confident in their correct labels than in their incorrect labels.

4.3.4.2 Recognizing Difficult Instances

In the preceding section, we had novice reviewers provide confidence scores for labels they had previously provided (i.e., the analysis was retrospective). We now focus on the prospective case, in which we give an inexperienced annotator the option of refusing to provide a label for difficult instances. We used two

datasets for our empirical evaluation: the COPD dataset (33) (Table 2.2) and Crohn's, which is from a systematic review of randomized controlled trials of monoclonal antibodies and other anti-TNF biologic agents for Crohn's disease. The datasets comprise 1606 and 2020 potentially eligible citations, respectively. For our 'gold standard' labels, we used labels provided by the expert reviewer who originally conducted the review. We then had an inexperienced reviewer screen both datasets, allowing her to refuse to label instances she thought difficult. We trained her in the standard way; for both datasets, an experienced reviewer familiar with the topic explained the inclusion criteria to her (i.e., what constitutes a 'relevant' study) and classified a few citations with her during a period of approximately thirty minutes. The inexperienced reviewer designated 9% (144 out of 1606) of the instances in the COPD dataset as difficult, and 6% (119 out of 2020) of those in the Crohn's dataset as difficult.

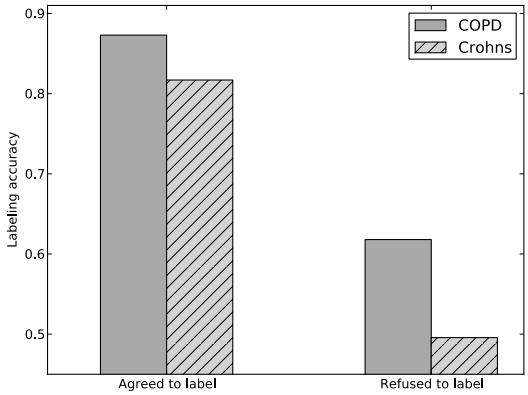


Figure 4.7: Novice reviewer labeling accuracy for those examples she was willing to label (left) and for those she designated as 'difficult' (right), over two datasets – COPD and Crohn's. See text for details.

We also asked the reviewer to label those instances she designated as difficult as best she could (i.e., we asked to which class she would assign an instance, were she forced to provide a label). We were thus able to compare the reviewer's accuracy over the examples she designated as difficult to her accuracy over the rest of the data. The hope is that the inexperienced labeler can categorize easy instances with high accuracy, while being able to recognize instances she is likely to mislabel. Thus we would expect her labeling performance over the two subsets of instances (difficult, easy) to conform to our expectations as modeled in Section 4.3.3.

This assumption is supported by Figure 4.7, which plots the novice labeler's accuracy over the instances she felt confident enough to label (left) and those that she designated as being difficult (right) over both datasets. For the COPD dataset, she achieved nearly 90% accuracy on the former set, and just over 60% accuracy on the latter. Similarly, on the Crohn's data she was 82% accurate on the instances she agreed to label, but performed poorly on those she refused to label ($\sim 50\%$). Reassuringly, this is in line with our modeling assumptions.

4.3.4.3 The Difficulty of Predicting Difficulty

In the proposed strategy we rely on novice labelers to inform us that instances are difficult. A natural alternative approach is to instead build a model which automatically identifies 'difficult' examples. This would allow us the same advantages – we could assign hard instances to experienced labelers and easy examples to novice labelers – while saving us some of the cost by reducing the number of (otherwise useless) 'difficult' labels provided by the inexperienced labeler(s).

The problem is that, at least in our application, predicting which instances labelers will designate as difficult is a non-trivial task. To investigate the feasibility of building such a predictive model, we first attempted to induce a standard Bag-of-Words (BOW) SVM over each of two systematic review datasets for which an inexperienced labeler indicated which instances were difficult. In ten-fold cross-validation, which is actually an optimistic assessment because in practice one would need to start using the model long before ninety percent of the data was labeled for it to be useful, predictive performance was quite bad. In particular, the model achieved an average sensitivity to difficulty examples of 64.3% with an average specificity of 53.6% on one dataset (COPD), and an average sensitivity of 66.2% with an average specificity of 49.2% on the other.[1]

Nor does it appear that model uncertainty correlates with human uncertainty. To show this, we induced an SVM over the entire dataset – again, this is therefore an optimistic assessment. As above, we used the 'difficult'/'not difficult' labels

[1]Here we used a linear kernel (and grid search to find the cost parameter c). Results using both an RBF and polynomial kernel were similar for both datasets.

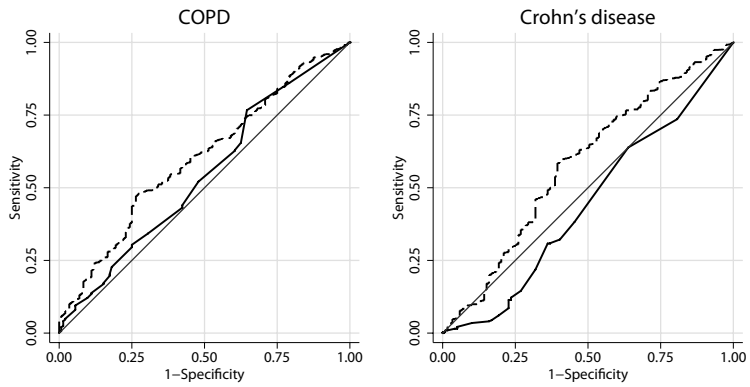


Figure 4.8: ROC curves showing discriminatory capability with respect to predicting which instances will be labeled 'difficult'. The dotted line corresponds to the distance to the hyperplane, and the bold, solid line to the feature-entropy score (Equation 4.7). Neither measure is particularly good at predicting difficulty.

provided by the novice reviewer as the target concept. We then generated a ROC curve using the ranked distances to the induced hyperplane, shown in Figure 4.8 as the dotted line. It is clear in the figure that model uncertainty is not a good predictor of human uncertainty (i.e., difficulty).

Human annotators obviously operate in a very different 'feature-space' than BOW classifiers. We have previously shown using labeled features (see Chapter 3) to be helpful in active learning (165) and could hypothesize that it is more realistic to base uncertainty on such information (e.g., words or n-grams associated with a specific polarity/class). Here we define a metric of uncertainty over labeled terms that scales the log term entropy in a document by the log of the total number of terms therein. Technically, denoting the number of positive terms in a given document as $\mathcal{T}^+$, the number of negative terms $\mathcal{T}^-$, and the total number of labeled terms in a document N, we have:

$$-\log(N) \cdot \log\left[\frac{\mathcal{T}^+}{N}\log\left(\frac{\mathcal{T}^+}{N}\right) + \frac{\mathcal{T}^-}{N}\log\left(\frac{\mathcal{T}^-}{N}\right)\right] \tag{4.7}$$

Intuitively, this feature-entropy score is a proxy for difficulty because it is large if there are many, conflicting terms in a document. Disappointingly, however, this measure fares worse than model uncertainty in its ability to discriminate 'difficult' from 'not difficult' examples, as shown in Figure 4.8.

While it would obviously be premature to conclude from this preliminary

evidence that automatically predicting which examples are difficult for use in instance allocation is impossible, it does demonstrate that the problem is non-trivial, and straight-forward techniques don't work. We therefore argue that reliance on novice experts to assess difficulty is an appropriate and effective strategy, at least for our domain.

4.3.5 Empirical Results – Citation Screening

We have demonstrated that our meta-cognitive MEAL strategy can be successful under certain conditions (Section 4.3.3), and that these assumptions hold in practice, at least in the context of biomedical citation screening (Section 4.3.4). Bringing these two points together, we now demonstrate the efficacy of our strategy for MEAL on datasets collected from our deployed biomedical citation screening system. We first outline our experimental setup, defining the actual cost structure in our application and discussing pertinent algorithmic details. We then show that under the presented cost model, our meta-cognitive MEAL outperforms baseline strategies over two citation screening datasets.

4.3.6 Experimental Setup

We ran experiments on the COPD and Crohn's datasets. In this case, two experts (one experienced and one novice) screened the citations comprising the datasets, deciding which were 'relevant' and which were 'irrelevant' to the review at hand. We again use the experienced expert's labels as the 'gold standard'.

As discussed above, evaluation over these datasets is somewhat complex, as one must realistically assess the trade-offs involved, as well as the total cost associated with different outcomes. For example, as mentioned previously, the cost structure here is asymmetric; 'false negatives' cost significantly more than 'false positives'. We thus use the weighted evaluation metric described by Equation 4.4, which expresses this trade-off. To recapitulate, we assume that sensitivity to the minority class of 'relevant' citations is λ times as important as mitigating cost; note that we assume cost is normalized).[1] We denote this 'utility' metric

[1] Here we are assuming that cost is $\propto$ *time* $\cdot$ *expertise*.

U_λ, quantifying this tradeoff with the λ parameter. Recall that this exercise resulted in a λ of 19 for the citation screening case.

$$U_\lambda = \frac{\lambda \cdot sensitivity + (1 - cost)}{\lambda + 1} \tag{4.8}$$

To estimate the costs involved with each MEAL strategy, we use a rough estimate of the salaries for the two reviewers which we converted into a unit cost (i.e., a cost per second). This allows us to calculate the cost of labeling given an estimate of per-citation annotation time. Here we make the simplifying assumption that all citations take thirty seconds to label, an empirical average taken over the dataset. We refer to the cost of acquiring a labeled training set with a MEAL strategy as the *upfront labeling cost*. Once this set has been collected there are two additional costs that must be taken into consideration.

First, some citations in the training set may have been mislabeled. The direction of this mislabeling has different costs associated with it; false positives will be subsequently retrieved in 'full text', which is quite expensive. In practice, all examples designated as 'relevant' by novices would be re-screened by the project lead (expert) in order to avoid incurring this cost unnecessarily. Therefore, we follow this procedure in our evaluation; we simulate the experienced expert re-screening all the documents designated as positive by the novice reviewer. False negatives are not directly accounted for in the cost model. Instead, we incorporate these into our evaluation by considering the overall sensitivity of a strategy. Thus in our case, sensitivity is calculated over both the training data (i.e., if a 'relevant' citation has been labeled as 'irrelevant' then sensitivity suffers) and over those instances classified by the induced model. This evaluation setup is appropriate for the finite pool scenarios we have previously discussed (Section 4.2; see also (168)). In such scenarios the primary aim is not to induce a good predictive model, but rather to categorize a fixed set of instances. The second additional cost involves those instances classified by the model induced over the acquired training set as 'relevant'. Every example that the model predicts as being 'relevant' must be screened; for this we charge the average cost of the two experts screening a citation. The instances the model designates as 'irrelevant'

are ignored, which may affect sensitivity. The above cost is summarized by the following equation:

$$cost = cost(\mathcal{L}) + cost(FP_h) + cost(TP_c + FP_c) \tag{4.9}$$

Where $\mathcal{L}$ refers to the cost of labeling data, FP_h to false positives due to human misclassifications, and TP_c, FP_c to instances correctly and incorrectly classified by the classifier as relevant (respectively).

We conducted experiments as follows. First, we gave each MEAL strategy two seed instances; one randomly selected from the set of 'relevant' instances and the other from the set of 'irrelevant' instances. We allowed each strategy to spend \$25 per round (iteration) on labeling.[1] Recall that a round is an iteration of active learning. After each round, we calculated the sensitivity (proportion of identified 'relevant' citations) and the total cost (as described above). We then combined these into a single metric that corresponds to the utility achieved for a given upfront labeling cost; U_{19}. All presented results are averages over ten independent runs in which each strategy received the same seed set selected for a given run.

4.3.7 Algorithmic Details

We make some small modifications to the ProActive (53) and meta-cognitive approaches for our empirical experiments. Both random baseline strategies are unchanged. As mentioned in Section 4.3.1.1, we used the first proposed variant of ProActive learning here, because it is a more natural fit for the data. In particular, examples that the novice reviewer labeled as 'difficult' were treated as instances this expert refused to label. We then induced a probabalistic classifier to predict the probability of a novice expert providing a label for a given example. We plugged this probability into Equation 4.5.

For meta-cognitive MEAL, we used the second variant (i.e., Algorithm 3). Because of the asymmetric cost structure, we decided it would be most advantageous for the experienced reviewer to double-check the novice's labels, even

[1] 25\$ was selected somewhat arbitrarily; we felt it provided an appropriate amount of granularity.

if there are no remaining instances that have been explicitly designated as difficult, i.e., if $|Q_e|$ is 0. Therefore, in line 10 in Algorithm 3, we set the bias to 1, implying that the experienced expert will re-label the instance in $\mathcal{L}$ classified by the novice that is least likely to be labeled correctly, according to the model induced over the experienced expert's training data. A caveat here is that because the experienced expert will review all instances that the weak expert labeled as 'relevant', we limit re-labeling during MEAL to instances that have been designated as 'irrelevant' by the novice. We set $\mathcal{W}$ to [.5, .5], thus enforcing an equal workload distribution.

4.3.8 Results

Figures 4.9 and 4.10 show the results over the COPD and Crohn's datasets, respectively. The most important observation is that the meta-cognitive MEAL approach consistently dominates the others, in terms of the metric of interest, U_{19}.

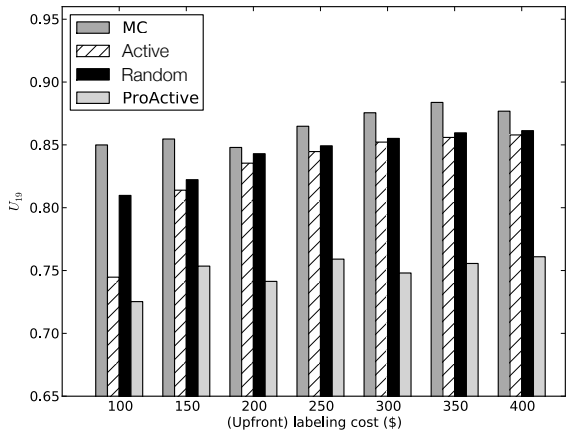


Figure 4.9: Upfront label cost versus U_{19}, Chronic Obstructive Pulmonary Disease (COPD).

ProActive learning fares poorly here. This is because it requests labels from the experienced expert almost exclusively, due to its greedy nature, and thus acquires relatively few (pricey) labels. Additionally, over both datasets the model induced to predict which instances the novice would likely refuse to label performed poorly, further hindering PAL's performance. The random strategies are relatively competitive with one another. Interestingly, the random strategy here

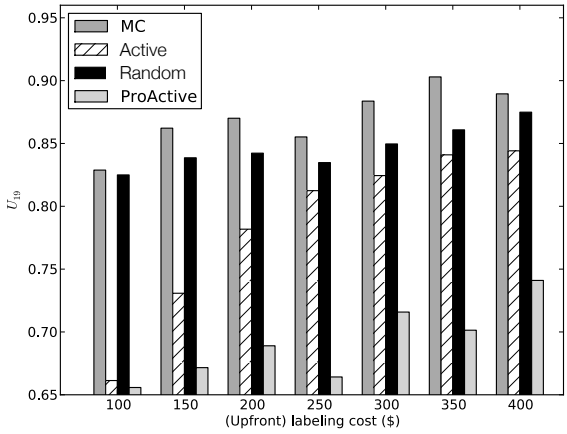


Figure 4.10: Upfront label cost versus U_{19}, Crohn's.

outperforms active random, contrary to the results over the sentiment analysis task. Indeed, this is in line with our previous observations that uncertainty-sampling active learning can perform poorly when there is significant class imbalance (165). While we wanted to focus on instance allocation in this work, and not the active learning scoring/VOI function, this suggests that perhaps combining the meta-cognitive approach with a different active learning criteria may perform even better (recall that we use standard uncertainty sampling to rank $\mathcal{U}$).

4.4 Modeling Annotation Time to Reduce Workload in Active Learning

We have thus far addressed one unrealistic assumption made in active learning work, namely that there is a single, infallible expert who provides labels. In this section we address a second naïve assumption often made in active learning: that the cost of acquiring labels for all instances is the same. We propose a regression model that predicts the time it will take to label an instance given its characteristics (e.g., length) and incorporate this prediction into the active learning query function, $\mathcal{Q}$. Normalizing the estimated utility of a given label in terms of a specific active learning criterion (e.g., version space reduction) by the expected cost of acquiring it effectively maximizes the return on investment (ROI) (72).

Using real data that we collected via our in-house annotation tool built for citation screening called *abstrackr* (described at length in Chapter 6), we demonstrate that this strategy outperforms more traditional 'greedy' active learning strategies, which tacitly assume a uniform per-instance labeling cost. In other words, when the predicted time to label an instance is factored into the decision of which examples to have the expert label, a better model can be induced in the same amount of time (i.e., at the same cost).

4.4.1 Modeling Experts

It has been shown that scaling the expected value of attaining a label for a particular instance by the cost, in terms of time, of acquiring said label can improve the performance of active learning (72, 146). However, deriving a statistical model to predict how long it will take to label a given example remains an open challenge (9). Indeed, Settles et al. demonstrated that in certain cases, *if* the true time to annotate were known *then* performance could be improved; however their model was inadequate in predicting labeling times on their dataset, and thus did not improve performance.

We hypothesized that, on average, annotation would take longer in the beginning of the screening process, while the reviewer familiarizes him or herself with the topic and screening criteria, and would gradually decrease thereafter. Previous work on predicting annotation times has not taken into account expert learning rates. Furthermore, in line with Settles et al. (146), we assume that longer documents would take longer to annotate. These assumptions were borne out by the empirical data collected from an ongoing citation screening project.

Figure 4.11 shows the relationship between mean annotation time and the order in which abstracts were reviewed. This relationship is shown in the smoothed dashed line, obtained from locally weighted linear regression with a sliding window of width 80% of the observations (lowess smoothing). The clear downward trend is intuitively agreeable; the annotator is learning as they label documents, and their speed thus increases as they become more familiar with the task. Moreover, as evidenced by the plot, their learning rate is more pronounced at the start of the task, and tapers off toward the end. There is also clear correlation be-

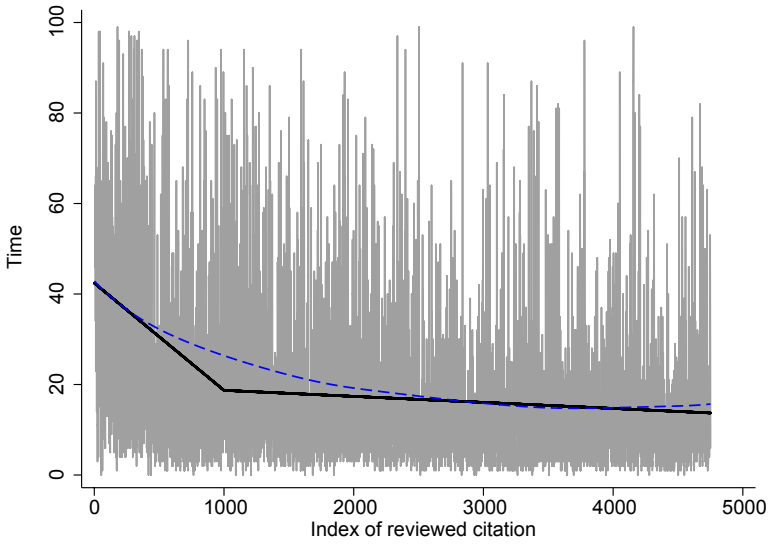


Figure 4.11: Document labeling time (in seconds) versus the order in which it was labeled. The dashed line shows a moving weighted average, the solid line two linear splines that captures this shape.

tween document length and labeling time, as can be seen in Figure 4.12, which plots the association between document length and annotation time and a line fit to this data (note the upward slope). In both plots, we do not show points for 106 documents (out of 4,751) that had associated labeling times longer than 100 seconds. These were considered outliers (it's likely that the reviewer became distracted while the tool was displaying these abstracts), and they made the plots difficult to read.

In addition to order and document length we also considered the correlation between model uncertainty, i.e., distance from the induced SVMs' hyperplane, and labeling time. It has been conjectured elsewhere that examples that the model is uncertain about may be in some sense difficult and thus take longer to label (53). To test this, we induced a model over all of the labeled data, and then computed the distance of each document to the separating hyperplane, a proxy for uncertainty (see Section 4.1). As shown in Figure 4.13, a correlation between model uncertainty and labeling time exists, but is rather weak compared to the observed correlation between, e.g., document length and labeling time. In particular, Spearman's correlation coefficient for the former is -0.05, whereas for the latter it is 0.39 (P-values < 0.001 for both). More problematically, the uncertainty will be extremely unstable at the start of active learning, as the

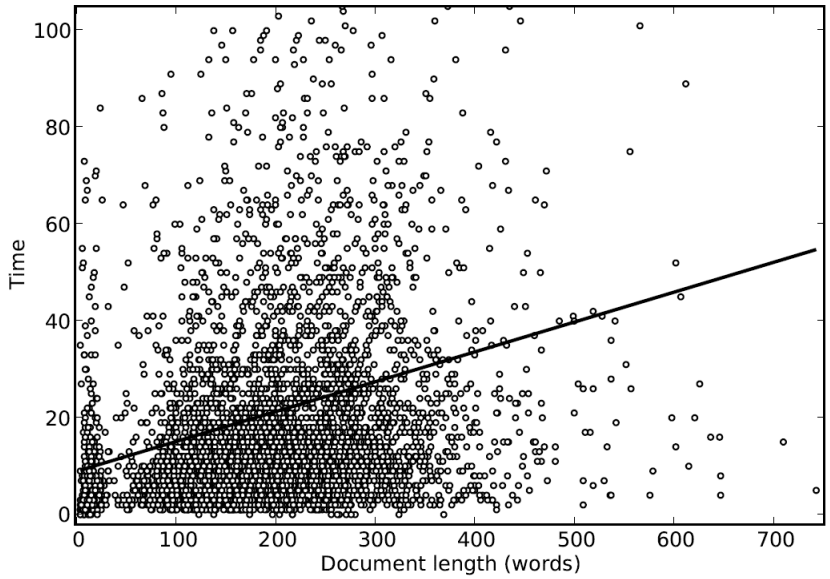


Figure 4.12: Document labeling time (in seconds) versus length (in words).

hyperplane will readjust dramatically as each new labeled example is acquired. For these reasons, we do not include the uncertainty in our annotation time prediction model.

We performed a regression analysis to predict the average time to annotate each abstract based on the order in which it is screened (i.e., first, second, n-th) and its length. We used a linear spline with a single knot at 1,000 abstracts to approximate the nonlinear relationship depicted by the solid line in Figure 4.11, as this seemed a natural way of modeling the learning curve. Using 1,000 documents for the spline regression was arbitrary; we just wanted to show that the learning rate increases rapidly at the start of active learning and more slowly thereafter. Linear mixed models with autocorrelated errors (to account for similarity of successive abstracts) and with information regarding which abstracts were screened in the same 'session' (to account for 'session'-specific effects) yielded very similar coefficients to those of an ordinary least squares regression, and we therefore used the latter model. Specifically, we model the time to screen a document d as follows:

$$\hat{y}_d(\beta) = \beta_0 + \beta_1 length(d) + \beta_1 n_1 + \beta_2 n_2 \tag{4.10}$$

where the n_1 and n_2 variables are functions of the number of documents that

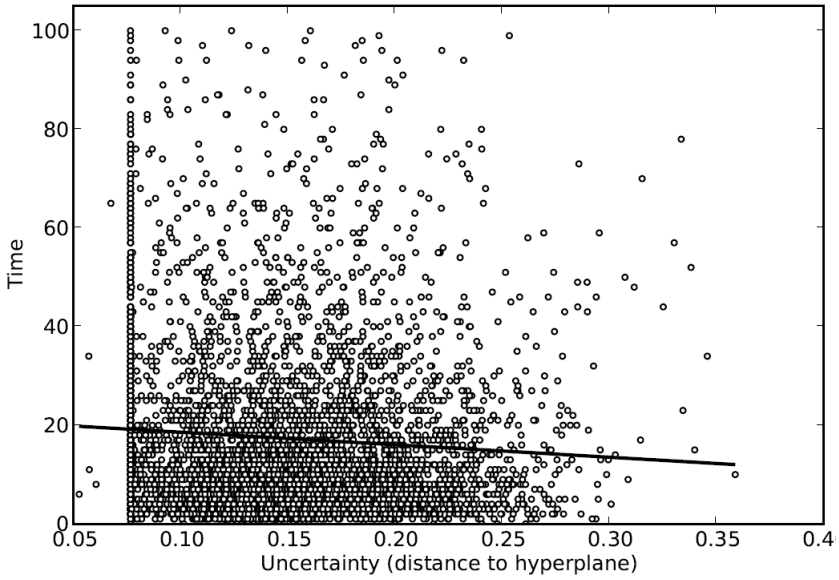


Figure 4.13: Document labeling time versus its distance to the hyperplane in an SVM induced over the entire dataset.

have already been labeled, which we will denote by n. Specifically, n_1 is n when fewer than 1,000 documents have been labeled, and fixed at 1,000 thereafter, while n_2 is 0 when fewer when 1,000 documents have been labeled and $n - 1000$ thereafter. This models the desired spline, which reflects the change in the annotator's learning rate.

Of course, while active learning is ongoing in practice, β is unknown. We therefore learn an approximation to β, $\hat{\beta}$, online using standard least-squares regression and the annotation times of the documents labeled thus far as target values. We then simply substitute $\hat{\beta}$s for the βs in Equation 4.10. See Algorithm 4 for more details.

4.4.2 Active Learning with Predicted Labeling Times

Our algorithm for active learning with predicted labeling times is shown in Algorithm 4. We first use a small sample of labeled data to get an initial estimate of the β coefficients. Additionally, we induce an initial hypothesis with which to begin active learning.

At each step in the active learning loop, which begins at line 5, we select for labeling the 'best-value' document, i.e., the document with the largest payoff per estimated time unit. This is shown in line 6, where d^* denotes the document selected for labeling by the reviewer. We then have the reviewer label this

document, and record the time it required (lines 7 and 8). Next, we re-train our classifier over the newly augmented training set (line 9). Finally, in line 10, we update our estimate of the β coefficients using the document labeling times observed thus far. In this way, we can estimate how long it will take to screen the remaining documents, given their length and the order in which they'll be screened, based on the times taken to screen the documents labeled thus far. This prediction is used as the denominator in line 6.

Algorithm 4 Active Learning with Labeling Times

Input: Learning algorithm $\mathcal{A}$, scoring function $\mathcal{Q}$, unlabeled dataset $\mathcal{U}$, labeled data sample $\mathcal{L}$, time budget $\mathcal{T}$

2: $t \leftarrow 0$
$\hat{\beta} \leftarrow$ least squares estimate using $\mathcal{L}$ {initial estimate of β coefficients}
4: $\hat{h}_t \leftarrow \mathcal{A}(\mathcal{L})$ {learn initial hypothesis}
while $t < \mathcal{T}$ **do**
6: $d^* \leftarrow \underset{d}{\text{argmax}} \frac{\mathcal{Q}(d)}{\hat{y}_d(\hat{\beta})}$ over $\mathcal{U}$
$\mathcal{L} \leftarrow \mathcal{L} \cup d^*$; $\mathcal{U} \leftarrow \mathcal{U} \backslash d^*$ {label selected point}
8: $t \leftarrow t +$ time taken to label d^*
$\hat{h}_t \leftarrow \mathcal{A}(\mathcal{L})$ {rebuild model}
10: $\hat{\beta} \leftarrow$ least-squares estimate using $\mathcal{L}$ {recompute estimate of β coefficients using labeled data}
end while

12: **Output:** Learned hypothesis h_t

4.4.3 Experimental Results

In this section, we turn our attention to an empirical evaluation of the proposed method. This is meant to demonstrate the advantage of taking into consideration the predicted time-to-label in selecting examples to have annotated in active learning.

To evaluate performance we use the weighted metric U_{19} (Equation 4.11, which is a slight modification of Equation 4.8), presented in Sections 4.2 and 4.3.5. Briefly, we are interested in two quantities; the burden imposed on reviewers and the number of relevant citations correctly identified. Previously, we have used the number of documents labeled as a measure of the former; here we use the actual labeling time, which we have collected. The sensitivity reflects the total proportion of 'relevant' instances in the pool identified, *taking into account the data with which the model was trained.* This is a subtle but important point:

for example, if an active learning querying strategy consistently selects for labeling relevant documents, it is effectively 'rewarded' for this behavior. Note that this is not the same as testing on training data; we do not attempt to predict the labels for documents included in the training set. Rather, we are quantifying the fraction of relevant documents correctly identified using a particular strategy, regardless of whether these documents were manually labeled relevant or were correctly predicted to be relevant by the classifier: this is appropriate for *finite-pool* scenarios. Note that without using any machine learning techniques, both the burden and sensitivity are 100%, as all relevant citations are identified, at the expense of the reviewers manually perusing all of the citations. Formally, we have:

$$U_\lambda = \frac{\lambda \cdot sensitivity + (1 - workload)}{\lambda + 1} \tag{4.11}$$

where the measure of the workload, i.e. the annotation time, is assumed to be normalized to fall in the range [0, 1]. We again use $\lambda = 19$, as was elicited from a project lead for a specific systematic review (Section 4.2; (165)).

We compare the strategy of taking into account the predicted time it will take label a document when selecting examples with a strong baseline strategy that we have previously shown to outperform random sampling (165). Figure 4.14 plots U_{19} against the number of instances labeled, i.e., the size of the training set used to induce the classifier. In this case, we quantify workload by the number of documents that must be screened by a reviewer. This includes the number of labeled documents and the number of documents predicted to be 'relevant' by the induced model, because these will need to be screened, whereas those documents that are designated 'irrelevant' by the classifier need not be screened. Given this result, we use our previously developed active learning strategy, rather than random, as our baseline.

To measure workload, we would like to use the actual time spent screening citations, rather than the raw number of documents screened. This is a bit tricky, however, because the time it will take to screen a particular citation is at least partially a function of the order in which it is screened (see Figure 4.11).

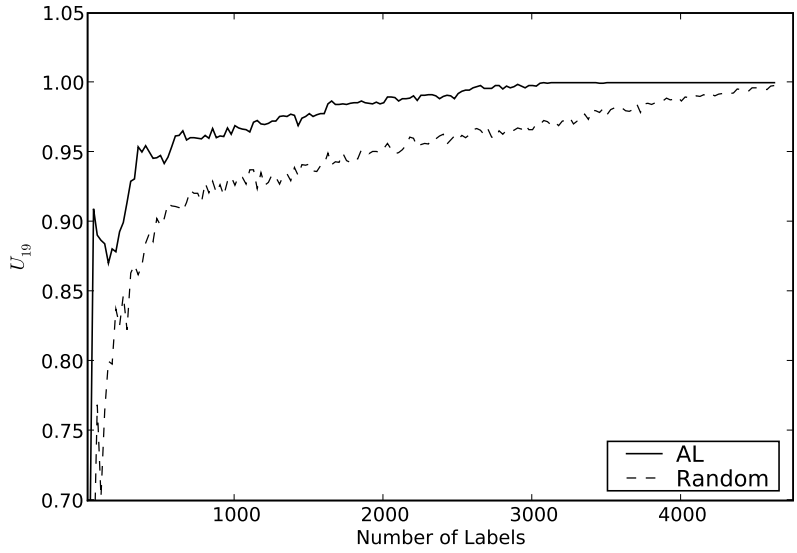


Figure 4.14: Classifier performance of active learning and passive learning.

Thus we cannot simply use the raw observed screening times in our experiments, because those times make sense only when the documents are labeled in the order in which the reviewer originally screened them. Therefore, to calculate the time spent labeling a citation for our experiments (line 8 in Algorithm 4) we use an order-adjusted time.

Denoting the raw observed time taken to label a document d by t_d, we have: $r_d = \hat{y}_d(\beta) - t_d$, where here we use the original order in which d was labeled for n (see Equation 4.10). Then r_d is the residual time taken to screen a citation, unaccounted for by our model. We then recompute $\hat{y}_d(\beta)$, setting n equal to the number of documents labeled thus far in the ongoing experiment, and subtract from $\hat{y}_d(\beta)$ the residual, r_d.

There is one more factor that complicates our evaluation; in addition to totaling the time spent labeling, we must take into account the amount of time it will take to label the documents that were predicted to be relevant. However, the 'true' annotation times for these documents will be partially contingent on the order in which they are screened. To eliminate this issue, we first sort all of the documents classified as 'relevant' by the model in descending order of length, and then simulate labeling them in this order. Finally, we compute a normalization constant for workload, because it is expected to fall between 0 and

1, as follows: sort all of the documents in descending order of document length, and sum the (simulated) time taken to label them in this order.

To recapitulate, we quantify performance using the U_λ metric, which is a weighted mean of the two quantities of interest: yield and burden. The former is the fraction of relevant citations correctly identified, the latter is a measure the total reviewer workload. In this case, we quantify workload by the total labeling time, which includes the time taken to label the training set, as well as the time taken to screen the citations categorized as 'relevant' by the classifier. The λ in this case was elicited from a reviewer, as we described above (see also (165)). A final note on evaluation: because we have extreme 'class imbalance', i.e., there are far fewer relevant than irrelevant citations, we under-sample the majority class of irrelevant citations before training our classifiers for evaluation. In other words, we remove irrelevant citations from the training set at random until there are an equal number of irrelevant and relevant citations. This strategy has been shown to be effective in mitigating the effects of class imbalance (see Chapter 2).[1]

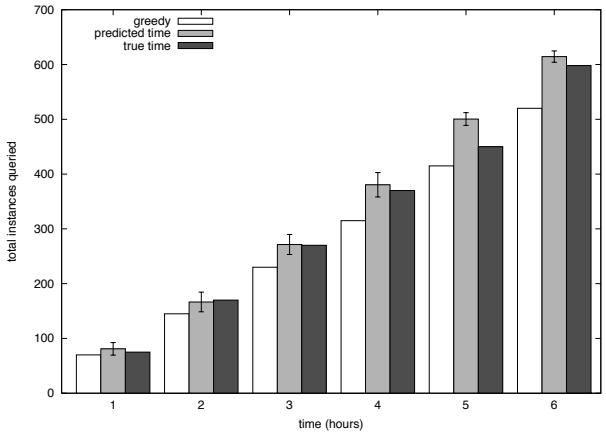
We compare three active learning strategies, described as follows:

- **greedy**: This strategy greedily selects for labeling the most promising document according to the dually-supervised active learning strategy we introduce in the following chapter (165). We use this as our 'base' active learning strategy for all three strategies, but any active learning strategy could be used in its place. The particular active learning criterion is not our focus here.

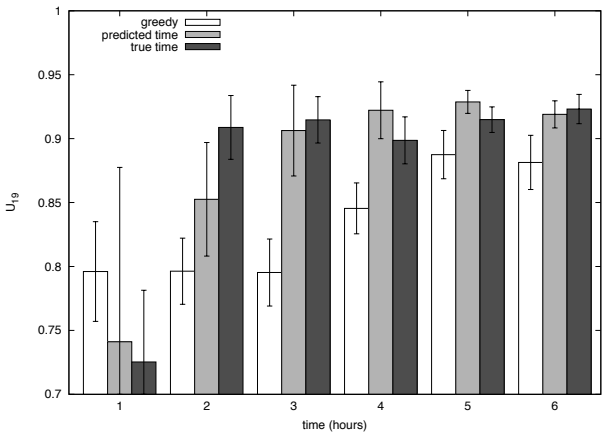
- **predicted time**: This method divides document scores (again a function of the labeled terms therein; see Chapter 5 for details) by the predicted time it is going to take to screen them, based on the regression model described in Section 4.4.1 and the current estimate of β, $\hat{\beta}$. This is the strategy we are proposing be used in practice.

- **true time**: This is the same strategy **predicted time**, except that it

[1] We did not bag here, as this work was done prior to our work regarding bagging classifiers induced over undersampled datasets. We are confident that all of the conclusions here hold in the bagged case, as well.



(a) Time versus the number of documents screened.



(b) Time versus U_{19}

Figure 4.15: Empirical results. In both plots, the white bar corresponds to the **greedy strategy**, the light grey bar to the **predicted time** strategy, which normalizes by the predicted time-to-label, and the dark grey bar to the **true time** strategy, which also normalizes by the predicted time-to-label, but uses the 'true' β coefficients in doing so (see text).

uses the true coefficients, β, as learned over the entire time series. This approach is therefore 'cheating', because it uses coefficients learned over data that wouldn't be available during active learning. The idea is to see how this compares to using the predicted time approach, which uses an estimate of β.

Note that all three strategies essentially follow Algorithm 4. The key difference is line 6; the **greedy** strategy does not normalize by anything, the **predicted time** strategy uses $\hat{\beta}$, as shown in the algorithm, while the **true time** variant uses β in the denominator.

We use the proton beam (157) dataset for experiments (Table 2.2). Recall that this dataset comprises 4,751 citations, of which 457 the reviewer labeled as

relevant, i.e., retrieved in full-text.[1] Unfortunately, this is the only dataset for which we currently have recorded screening times, and thus is the only dataset we run experiments over.[2]

Our experiments were conducted as follows. We allotted six hours for (simulated) labeling, and evaluated performance every hour. We take the most recently reported performance at each check-in point, i.e., on the hour. All results are averaged over ten independent runs in this way. This experimental framework matches our scenario: we are assuming that we have a fixed amount of time to annotate a corpus, and want to evaluate our performance with respect to categorizing this set of documents under the time (equivalently, budget) constraints.

Figure 4.15a plots the average cumulative number of examples that were labeled using each of the three strategies at the end of each hour. The error bars for the **predicted time** strategy show the standard deviations at each time point; the other two querying strategies are deterministic. It is reassuring that both strategies that take time into consideration are indeed able to have the reviewer label more citations in the same amount of time, compared to the **greedy** strategy. Interestingly, using the **predicted time** approach often results in acquiring more labels than when the **true time** strategy is used. We suspect that this is because the time prediction model learned online is 'pessimistic', in that it tends to predict that documents will take longer to label than they actually do. This is likely because of bias in the documents for which labels are requested during active learning (over which the time prediction model is subsequently induced); these tend to be difficult, and thus the 'true' labeling time is higher than it would be if an i.i.d. sample were used.

The average performances of the respective strategies at each time point are shown in Figure 4.15b. The error bars are standard deviations. Note that even

[1] We have previously used this dataset with labels from a different reviewer, who screened this data before the ABSTRACKR tool was developed. We had a colleague re-screen them in order to test our tool; the class distribution breakdown is thus slightly different in this case than in our previous work.

[2] Technically, we have additional datasets that have been collected via the *abstrackr* system introduced in Chapter 6, but we are not using these in methodological work because we want to eventually perform a large-scale empirical analysis over these datasets, and to ensure validity it is important that the datasets involved in this evaluation are not used during development.

the deterministic querying strategies have standard deviations because we have to under-sample the majority class (irrelevant citations) to mitigate the effects of the severe class imbalance, as described above. The first thing to note is that both strategies that take time into account outperform the **greedy** strategy at all points after the first hour. It is intuitive that taking the 'long-view' strategy should only pay off after some sufficient amount of time has passed. The **greedy** strategy (almost by definition) will rapidly achieve good performance, but will quickly exhaust its budget. On the other hand, time-sensitive strategies pay off by being prudent in their example selection; the aggregate benefit of this strategy takes some time to manifest.

It is also encouraging that our **predicted time** strategy, which learns to predict how long it's going to take to label citations online (i.e., during active learning), performs comparably to the **true time** strategy, which uses the 'true' model coefficients β, as learned over the entire labeled dataset. This is in contrast to previous work (146) in which the predictive model was not sufficiently accurate to achieve the same performance as when the true times were used. It is possible that our incorporation of the annotator learning rate, i.e., the number of documents labeled prior to the document for which labeling time is to be predicted, accounts for the success of our approach.

4.5 Conclusions

We have presented novel active learning methods for real-world scenarios that relax the unrealistic assumptions often made in active learning. Such methods are necessary if active learning is to be useful in real-world tasks.

More specifically, in Section 4.3, we presented the problem of Multiple Expert Active Learning (MEAL) and outlined the difficulties therein. We presented a novel strategy for MEAL that relies on the participating novice labelers to indicate which examples are difficult, allowing the strategy to best exploit experienced (and expensive) labelers. Further motivating this approach, we provided preliminary evidence that automatically predicting which instances are difficult is a hard task. Moreover, we provided evidence that novice reviewers

have the necessary meta-cognitive skills to assess which instances they are likely to mislabel. Our meta-cognitive strategy out-performed strong baselines, including a previously proposed approach to MEAL, on both sentiment analysis and biomedical citation screening tasks.

We proposed a model for predicting the time it will take an expert to label a given example. We showed that taking this time into consideration during active learning can improve performance. Specifically, we demonstrated that normalizing the active learning score assigned to an instance by the predicted time it will take to label it results in a better performing system. We presented a simple spline regression that incorporates document length and the order in which a document is labeled as predictive variables. The spline serves as a simple model for the annotator's learning rate. The coefficients for this model can be learned online, as active learning is ongoing. We showed that using this 'return on investment' approach results in better performance in the same amount of time, compared with the commonly used greedy active learning strategy.

5

Dually Supervised Active Learning

In Chapter 3, we introduced the dual supervision paradigm, which exploits labeled features (e.g., in text classification, words or n-grams associated with a specific class) in addition to standard instance labels. In Chapter 4 we then reviewed the active learning paradigm and proposed methods to extend the framework for more realistic scenarios. In this chapter we combine these two approaches by exploiting labeled features during active learning. In particular, we use the external knowledge captured by such features to guide the active learning process, specifically by using them to inform the active learning scoring function $\mathcal{Q}$. Dual supervision naturally lends itself to interactive techniques: experts might, for example, want to 'tell' the model about words that seem to be confusing the classifier, or they may want to provide Information Retrieval-style feedback regarding the class of primary interest.

There is an additional advantage to guiding active learning with labeled features in the context of imbalanced datasets. Because we assume that the experts know terms that are associated with the minority class *a priori*, this knowledge is external to the points selected for labeling thus far. Exploiting this supervision to inform for which instances labels are requested may thus mitigate the effects of the sample selection bias inherent to active learning. In particular, this external knowledge may help to discover disjunctive sets of minority instances, thereby sidestepping the 'missed-cluster' effect (139, 165). We will revisit this

issue in Section 5.2. We note that parts of this chapter appeared in the *Proceedings of the 16th ACM SIGKDD conference on Knowledge Discovery and Data mining* (KDD 2010) (165).

5.1 Related Work

Before developing our own dually-supervised active learning method later in this chapter, we first review existing strategies in this vein.

5.1.1 AL with Labeled Features

We first review work on active learning with labeled features *only*, i.e., without any instance labels. Extending the generalized expectation (GE) framework for learning from labeled features, Druck et al. (58) proposed an active learning strategy for labeling features. Generalized expectation criteria – discussed in Section 3.1 – is a framework for incorporating arbitrary prior expectations into parameter estimation (112). Druck et al. (58) developed a pool-based feature approach to selecting features for an expert to label, analogous to a standard pool-based AL algorithm selecting instances for labeling.

In conjunction with GE criteria, they use a conditional random field (CRF) (95) as their underlying probabilistic model, though any generative model may be used in its place. They estimate the parameters of the CRF in a semi-supervised way that incorporates the provided feature information, as proposed by Mann et al. (109). As in (57), they add constraints to reflect the distance (KL divergence) between the current model's predicted feature-label distributions and the *a priori* expected distributions, as provided by the expert. In standard active learning one must specify the $\mathcal{Q}$ function that effectively ranks unlabeled instances with respect to the benefit expected should their labels be acquired; In Druck et al.'s case this query function performs a similar ranking over features, rather than instances.

Druck et al. (58) consider a few such feature-query functions. They emphasize that this task of selecting features for the expert to label differs from that of feature selection: in the former one looks to select features for which expert

feedback will assist the model, whereas in the latter one is specifying the features to be included in the model. Note also that a necessary extension of the pool-based framework for feature feedback is a *skip* option – essentially allowing the user to label a feature as being uninformative – because many features will have no meaningful class association. Druck et al. propose a few different uncertainty sampling based strategies wherein the learner attempts to pick for labeling the feature about which feedback will provide the largest decrease in model uncertainty. One could pick the feature with the highest expected information gain directly, but estimating this is computationally intractable, as it would involve computing expectations over all instances for all features. Instead, the authors propose scoring features as a function of the Total Uncertainty (TU), defined as follows:

$$\phi_{TU}(f_k) = \sum_i \sum_j f_k(x_i, j) H(p(y_j|x_i;\theta)) \tag{5.1}$$

where H denotes entropy, i and j index instances x_i and classes y_j,[1] respectively, θ is a vector of model parameters and the query is with respect to a feature k. Further, $f_k(x_i, j)$ is an indicator function which is 1 if feature k is in instance x_i at position j. The problem with this query function is that it will disproportionally select features whose values frequently assume **1**, i.e., those that often appear in instances. To mitigate this problem, Druck et al. (58) propose scaling ϕ_{TU} by the corresponding feature's count, C_k. However, they note that just dividing ϕ_{TU} by C_k would result in the opposite: the function would always select rarely occurring features. They thus propose the compromising heuristic shown in Equation 5.2, which they call Weighted Uncertainty (WU).

$$\phi_{WU} = log(C_k)\frac{\phi_{TU}(f_k)}{C_k} \tag{5.2}$$

Druck et al. (58) proposed a few other query function as well, but WU is the best performing of the bunch. Their method outperforms passive learning with features and traditional active learning over instances in simulated user

[1]Recall that here y_j is a *feature*, rather than instance, label.

experiments (i.e., using a feature oracle) over two sequence labeling tasks.[1] They also experimented with real users, and introduced a novel grid interface for labeling features. Notably, they show that feature annotations are cheaper than instance annotations. Our scenario is different from the one explored in the GEC case: we assume experts know beforehand a list of features that correlate with classes. Thus we are uninterested in requesting labels on particular features during AL. Furthermore, we would like to exploit features and instances dually, as opposed to the approach delineated above, which learns from features alone. The framework of GEC is general enough to accommodate dual supervision with a suitable specification, but we did not pursue this route. We now turn to methods more relevant to our scenario, i.e., dually-supervised methods that simultaneously exploit instance- and feature-labels.

5.1.2 Dually Supervised AL

In early work on active dual supervision, Godbole et al. (69) emphasized the human-computer interaction aspect of the process (69). They exploit term-level labels by adding single-feature psuedo-instances, similar to Raghavan et al.'s approach (130, 131). They present terms likely to be correlated with a given class to users interactively, allowing them to edit these if they so choose. In similar work, Stumpf et al. (152) carried out extensive user studies investigating 'rich' forms of user interaction with machine learning systems. In both cases allowing humans to impart supervision beyond instance labels substantially improved system performance, even when simple dual methods were used.

Raghavan et al. (131) outlined an augmented general active learning protocol that allows the model to select features (in addition to instances) for the expert to label during each iteration of AL. They then incorporated this elicited domain knowledge into the model during training. They first experimented with an idealized *feature-oracle* over benchmark text classification corpora. This allowed them to assess whether feature-feedback is at least theoretically helpful. They allowed the feature-oracle to pick the best k features, with respect to information

[1] Sequence labeling tasks are structured learning problems in which instances come sequentially; they are common in natural language processing and genetics, for example.

gain, at the outset of AL and then performed traditional uncertainty sampling over the pruned k-dimensional labeled feature space.[1] Their findings confirm that labeled features indeed can improve model performance; AL over the pruned space outperformed traditional uncertainty sampling over the original un-pruned feature space on five benchmark datasets.

Raghavan et al. (131) next investigated whether human labelers could, in practice, approximate the feature-oracle, thereby achieving similar gains. To this end, they obtained labeled features from human experts *a priori* (i.e., non-interactively, at the start of AL) with which to experiment. They simulated interactive feature feedback using these feature relevance labels during AL. They called this procedure of issuing label requests for both features and instances *tandem learning*. These labeled terms were then incorporated via a simple feature scaling technique. Specifically, they scaled the labeled features by an order of magnitude (10x) relative to the unlabeled features in the vectors representing each instance. Their experiments with this simulated 'human-in-the-loop' setup were promising; they achieved results comparable to the performance observed using the feature-oracle methodology. Raghavan et al. have since proposed additional tandem learning methods (130).

Attenberg et al. (11, 12) proposed a general framework for the task of selective data acquisition. They built upon the Pooling Multinomials method reviewed in Section 3.1, but the proposed methods would conceivably work with any dually supervised learner. The basic idea is to interleave requests for feature and instance labels, picking both cleverly. They first consider the obvious analogy to uncertainty sampling as applied to features. This method would request labels for features about whose class association it is least certain. However, Attenberg et al. (11) point out that this results in querying the expert for labels on noisy, non-discriminative features, thus wasting valuable expert time. Somewhat counter-intuitively, they go on to demonstrate that one of the most effective feature-querying strategies is to request labels for those features whose polarity the model is *most* certain about, the exact opposite of uncertainty sam-

[1] They also experimented with conducting the feature queries *after* AL was finished, with comparable results.

pling. The intuition is that this works because the expert will likely label the feature/word as being indicative of a class, whereas in the case of noisy features, e.g. 'the', the expert simply labels them as uninformative. They also propose a querying method based on the expected utility of a given feature, in terms of model improvement. They demonstrated that this utility method outperforms baseline feature-querying strategies.

In a similar vein, Liang et al. (101) proposed a unified perspective for acquiring heterogeneous forms of expert supervision. More specifically, they investigated techniques for simultaneously exploiting expert-provided constraints and standard instance labels. The measurements model is similar to the Generalized Expectation Criteria method (57) in that it learns from probabilistic expectations provided by the expert. The model cannot learn directly from discrete feature labels, though one could conceivably coerce labeled features into reasonable probabilistic constraints, as has been done in the case of GEC (57).

To unify these disparate forms of supervision, Liang et al. introduce the *measurements* abstraction. A measurement is an expectation of a function defined over the outputs of the unlabeled examples. For example, a measurement may be a fully labeled example, a partially labeled example, or a constraint reflecting a feature-label relationship. Their proposed Bayesian model then exploits all measurements, including instance-labels, by maximizing the posterior probability with respect to these. Starting with this as a learning model, Liang et al. (101) then addressed the question of efficient acquisition of additional supervision. They proposed a decision-theoretic approach, in which measurements that maximize expected utility are taken at each step. Roughly, utility here is the expected improvement in model performance achieved by acquiring a measurement less the cost of taking it.

We have summarized existing work on dually-supervised interactive learning. We propose our own strategy for this, CoFeature, in Section 5.3. But we first pause to consider why and when feature-level supervision might help inform active learning, particularly in imbalanced scenarios.

5.2 Hasty Generalization, or, When Might Dual Supervision Improve AL?

As discussed in Chapter 4, the citation screening task naturally fits within the pool-based active learning paradigm, in which the model requests labels for the unlabeled examples likely to be most helpful in learning the target concept. We initially experimented with active learning over a few citation corpora from previously conducted systematic reviews (Table 2.2), using standard uncertainty sampling methods with SVMs (158). In these experiments, uncertainty sampling resulted in models with high accuracy but poor sensitivity, compared to models trained on randomly selected data (168). This is obviously undesirable given the cost asymmetry present in the citation screening task discussed at length in the preceding chapters. In this secction we address the question of why uncertainty sampling might induce models with poorer sensitivity. We then discuss how this problem can be mitigated by exploiting labeled features to guide AL.

Uncertainty sampling methods focus on refining the current decision boundary (122). The idea is to first establish a rough approximation to the ideal decision boundary and then sequentially requesting labels for examples nearest it. Intuitively, this strategy exploits the labeler by ignoring examples whose labels are unlikely to move the decision boundary, thus expediting the training process. However, this strategy implicitly assumes that the initial approximation to the decision boundary is reasonable in the sense that as the learner continues requesting labels, the learned boundary will approach the optimal boundary. This assumption is violated in the case of disjunctive concept clusters (16, 139), as uncertainty sampling may continue to request labels along the initially discovered boundary, ignoring as-yet undiscovered partitions. We call this the problem of hasty generalization.

The most relevant existing work with respect to addressing hasty generalization is that of Schütze et al. (139), in which they discuss practical issues in active learning for text classification. They observed a phenomenon similar to that just described, which they referred to as the *missed-cluster effect*. They found that this was problematic in real world active learning for text classifica-

tion, particularly when there is class imbalance, confirming our own independent observations.

Other work (16, 122) has addressed this problem more generally as a trade-off between exploration (e.g., random sampling or via the Kernel Farthest-First heuristic (16)) and exploitation (e.g., uncertainty sampling) during active learning. The strategies proposed in these works decide at each iteration in AL whether to explore the space or to exploit what is already known. Typically this decision is taken stochastically, with the respective options weighted by the estimated likelihood of their being fruitful based on previous decisions. Thus at the outset of AL we are more likely to explore at each step, whereas once a large amount of training data has been acquired we are more likely to exploit. This approach fits naturally in the one-armed bandit framework, in which we are to select an arm to pull at each step that will maximize pay-off (16). The problem with such explore/exploit approaches in the more specific case of imbalanced data is that they are greedy insofar as they explore with probability proportional to how successful exploration has been thus far. These methods therefore tend to regress to standard active learning in the case of imbalance, because exploration will only rarely be fruitful, specifically on those rare occasions that a minority-class instance is selected. We note that He (78) has also recently proposed exploiting labeled features for rare class detection.

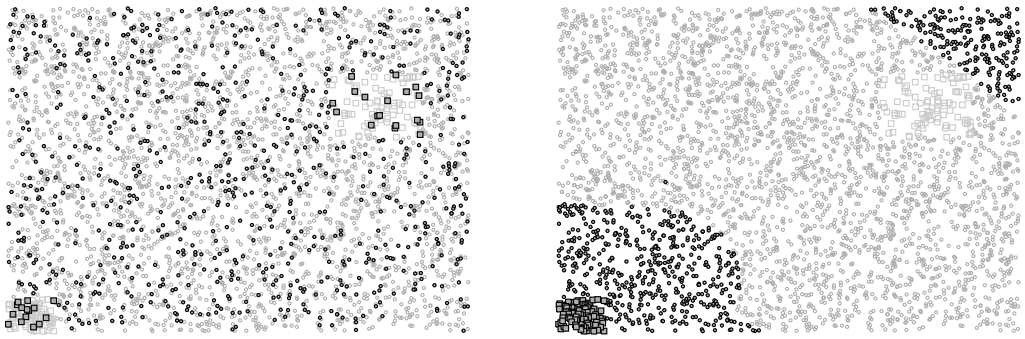


Figure 5.1: The left and right figures show the examples for which the random sampling and Simple (see 4.1) strategies requested labels, respectively. In both plots the entire pool of examples ($\mathcal{U}$, at the start of active learning) is shown; examples that are darkened are those for which a label was requested by the corresponding learning algorithm.

The problem of hasty generalization is perhaps easiest understood with a

toy example. Consider the two-dimensional target concept depicted in Figure 5.1. Here the instances represented by squares comprise the minority class, of which there are two clusters (one in the lower left-hand corner, the other in the upper-right quadrant). We simulated AL over this data using an SVM with an RBF kernel and two different learning strategies: passive, which randomly selects examples from $\mathcal{U}$ for the expert to label, and uncertainty sampling via Simple (158). The examples selected for labeling by these two algorithms are darkened in the two sub-plots, Figures 5.1a and 5.1b, which correspond to random sampling and Simple, respectively. We allowed the learners to request labels for 25% of the total data.

Figure 5.1a shows the examples that were selected using the random sampling strategy. In this case, the learner was trained on a representative, i.i.d. sample of the data, and discovered examples from each of the two minority clusters. However, random sampling was clearly inefficient, in the sense that it queried for the labels of many irrelevant examples, thus wasting our simulated expert's time. To expedite the training process, and to induce a more accurate model, one might appeal to uncertainty sampling here. But hasty generalization is a potential pitfall in this approach. This is illustrated in Figure 5.1b, which shows the examples for which Simple requested labels. The training examples selected via uncertainty sampling are visibly biased, clustering around the initial approximation to the decision boundary in the lower left quadrant. The learner completely misses the upper-right cluster of squares.[1] The active learner hastily generalized from the examples it initially encountered, and will subsequently misclassify squares in the missed cluster as circles.

The question, then, is: how can we exploit the expert via AL when we have an imbalanced class distribution and asymmetric costs? In the following section, we propose using labeled features to achieve this aim. In particular, labeled features – n-grams, in the case of text classification – that are known to the expert at the outset of AL can be used to circumvent the problem of hasty generalization by combining *a priori* knowledge with the model induced over the

[1]Note that labels for some circles (but no squares) in the upper-right hand corner were requested due to our use of the RBF kernel.

current set of labeled instances. Indeed, Shütze et al. (139) explicitly suggested that using domain knowledge may be a fruitful way of avoiding the missed-cluster effect. Consider the simple example above. If the expert *knows* that minority instances should exist somewhere in an upper right region of the feature-space (the interpretation of this, of course, would depend on the semantics of the features in the given task) then this could be used to guide the model to discover both clusters of minority instances.

5.3 *CoFeature*: A Co-Testing Approach to Dually Supervised AL

We now present a novel active learning strategy that exploits domain knowledge provided by the expert in the form of labeled features. We extend this model via a variant of the CW-SVM presented in Chapter 3 for situations in which the expert can provide ranked labeled features. We show that our methods outperform existing AL strategies on three systematic review datasets.

One way of looking at labeled features is as a distinct *view* of the data. A view is a particular feature space used to represent a given dataset. Blum and Mitchell (25) demonstrated that multiple, redundant views can be exploited in supervised learning through the *Co-Training* paradigm. Muslea et al. (119) extended this method for active learning via their *Co-Testing* strategy, which works as follows. Suppose we have two views, V_1 and V_2. Learn two hypotheses H_1 and H_2 over these views, respectively. Now define *contention points* as those unlabeled examples about whose labels H_1 and H_2 disagree, and request a label for one of these points. This approach is appealing because if these two models disagree on a particular example x, then by definition the label for x must be informative, as at least one of the two models is currently incorrect. Note that Co-Testing is a specific case of Query by Committee (66), reviewed in detail in Section 4.1.

We propose building a simple odds-ratio model over the expert-provided labeled features in tandem with a linear-kernel SVM over a standard bag-of-words (BOW) representation of the corpus. For the odds-ratio model, we use an

'odds-ratio' model calculated over labeled term counts, i.e., the ratio of positive to negative terms in a document. In particular, suppose we have a set of positive features (i.e., n-grams indicative of relevance), $\boldsymbol{\alpha}$, and a set of negative features $\boldsymbol{\beta}$. Then, given a document d to classify, we can compute a coarse likelihood of d belonging to the positive class as:

$$\frac{\sum_{w^+ \in \boldsymbol{\alpha}} I_d(w^+) + 1}{\sum_{w^- \in \boldsymbol{\beta}} I_d(w^-) + 1} \tag{5.3}$$

where $I_d(w)$ is indicator function which is 1 if w is in d and 0 otherwise. Note that we add pseudo-counts to both the negative and positive sums to avoid division by zero. The direction of this ratio gives a class prediction and the magnitude of the ratio gives a confidence.[1] For example, if d contains ten times as many positive terms as it does negative terms, the class prediction is + and a proxy for our confidence is 10.

We can now use this model for Co-Testing as follows. First, generate the set of contention points, i.e., those unlabeled examples about whose class membership the SVM model induced over the BOW representation disagrees with the labeled feature classifier defined above. Of these, select for labeling the example x with the largest ratio. In this case the SVM model predicts that x belongs to one class, but the labeled features present in x strongly suggest that it belongs to the other. The hope is that such examples will be informative to the model, given the disparity between the shallow "semantic" classifier that uses labeled features and the more nuanced "black-box" SVM method induced over the instances labeled thus far. Hopefully this strategy will avoid the problem of hasty generalization because it relies on information external to the current SVM model to select which instances are to be labeled (of course, this will depend on the structure of the minority points and the labeled-feature information provided). Once there are no contention points remaining, we may fall back on standard uncertainty sampling; at this point the SVM has likely acquired training data from – or else

[1] To ensure that the magnitude is symmetric in the respective directions, one may either flip the ratio such that the numerator is always larger than the denominator, or one may take the log of the ratio.

is already correctly classifying – points comprising clusters of minority instances, assuming the expert provided labeled features that correspond to said clusters.

The simple odds-ratio style model presented above assumes that the model has access to only binary feature-labels. But, as discussed in Chapter 3, in many cases the expert may also be able to provide a ranking over features, specifying which are more or less representative of class membership, relative to one another. Encoding such fine-grained information is an attractive proposition because it exploits the domain knowledge provided by the expert to induce a better generalized model, again thereby hopefully thwarting the aforementioned problem of hasty generalization. We can exploit such ranked-feature information via the constrained-weight SVM formulated in Chapter 3.

More specifically, we can use a CW-SVM (induced over ranked features and the instances in $\mathcal{L}$) as the second model in the Co-Testing framework. Note that in both cases we use standard SVM as V_1, i.e., the classifier ultimately responsible for making predictions. To use the CW-SVM, we must specify functional constraints between features of different ranks (see Section 3.2.2.4). Here we assume that the magnitude of the parameters associated with labeled features grows exponentially with their rank as the following Equation

$$f(x, y) = e^{-\kappa x} - e^{-\kappa y} \tag{5.4}$$

where x and y are adjacent ranks and κ is a parameter reflecting the magnitude of separation we expect between ranks. The intuition behind using this exponential function is that the presence of the highest ranked terms in a document are significantly more indicative of its relevance (or irrelevance) than lower ranking terms. This is thus the exponential variant of the CW-SVM defined in the Quadratic Program specified by Equations 3.21 through 3.25. The assumption to use the exponential ranks was made in part due to informal discussions with the our expert regarding the relative importance (in his view) of certain terms versus others. As in the simpler version of the CoFeature method, this classifier is used as a view to select contention points with the standard SVM model, i.e., as V_2.

5.4 Experimental Results

We first present experimental results using the simple odds-ratio based Co-Testing approach proposed above, which we will refer to as just *CoFeature*. We run experiments over three systematic reviews for which we were given labeled terms by the reviewers. These datasets are summarized in Table 2.2. We compare our approach to random sampling, uncertainty sampling via Simple, and uncertainty sampling over the labeled-features space as proposed by Raghavan et al. (131). We also present results using the CW-SVM method as the second view, rather than the odds-ratio model, over the proton beam dataset.[1]

5.4.1 Experimental Setup

As before, evaluation is carried out with respect to the metric of interest, i.e., U_{19} (for details see 4.2). Briefly, this metric emphasizes sensitivity to the minority class of "relevant" citations, as is appropriate in our scenario. We note that Simple outperforms our method on all datasets with respect to accuracy, but this is essentially meaningless in our scenario due to the low prevalence of the relevant instances. This again illustrates the necessity of using the appropriate metric for the task in evaluation.

All classification is performed using SVMs with linear kernels, as they have been shown to perform well over high dimensional text data (84). All SVMs are induced over a feature space comprising a binary bag-of-words encoding of concatenated citation title, keywords and abstract text, save for the method of Raghavan et al., which operates exclusively in the labeled terms space only. Evaluation is assessed over the as-yet unlabeled instances remaining in $\mathcal{U}$, as described in Section 4.2.[2]

Our experimental setup is as follows. We instantiate the four learners and give each of them labels for the same two 'seed' citations; one "relevant" and one "irrelevant". We then allow each learner to request five labels per round

[1] We only had expert-provided ranked labeled features for this dataset at the time the experiments were conducted.

[2] We tune the C parameter via grid-search prior to evaluation over the training data. We modify the search criteria to reward good performance with respect to both sensitivity and specificity, rather than overall accuracy, for all learners.

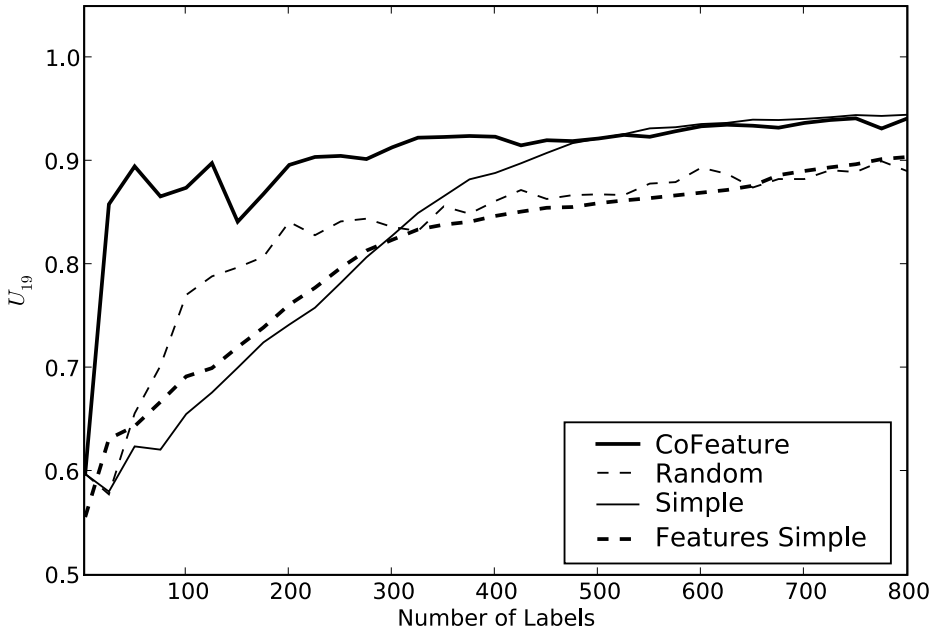


Figure 5.2: U_{19} over the COPD dataset. Our CoFeature approach outperforms all baseline methods.

of active learning. Every 25 labels, we evaluate the learners as described above and report results. Due to the severe class imbalance in our task, we undersample the majority class (at random) to make the class distribution uniform prior to building the classifier used in evaluation: see Chapter 2 for an in-depth discussion of this issue.[1] All reported results are averages over ten independent runs.

5.4.2 CoFeature Results

Results over the COPD dataset are shown in Figure 5.2. Recall from Table 2.2 that COPD is a smaller dataset than proton beam, comprising 1,601 citations, 196 of which are relevant. We show performance for up to 800 labeled training examples. We were given 22 labeled n-grams, fifteen positive and seven negative. Our CoFeature method maintains higher U_{19} until about the 500 label mark, at which point Simple performs comparably.

Figure 5.3 displays results over the micronutrients dataset. There are 4,010 citations in this dataset, 258 of which were found to be *relevant*. This is an inter-

[1]We do not bag here because these experiments pre-date our work on bagging learners induced over balanced datasets. But because bagging is primarily a variance-reduction strategy, we are confident that the conclusions drawn hold for ensembles of undersampled classifiers, as well, since these are based on point estimates.

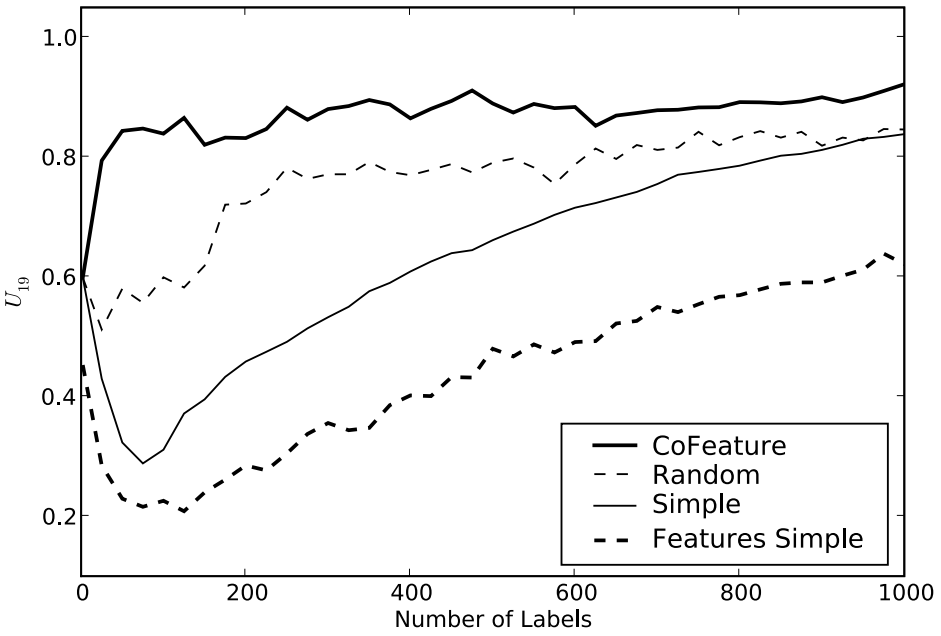


Figure 5.3: U_{19} over the micronutrients dataset. Our CoFeature approach outperforms all baseline methods.

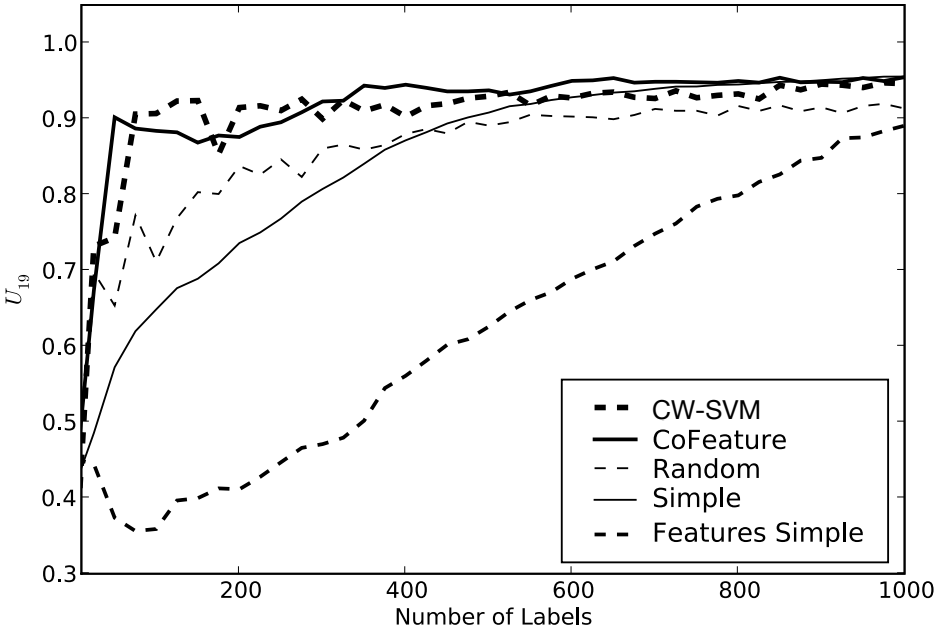


Figure 5.4: U_{19} over the proton beam dataset. Our CoFeature (and CW-SVM) approaches outperform all baseline methods.

esting dataset because the expert provided a preponderance of positive n-grams: 47 versus only two negative terms. The CoFeature strategy again dominates the other methods.

Figure 5.4 shows results over the proton beam dataset: these include results using Co-Testing with the CW-SVM method, as well. As shown in Table 2.2, there are 4,751 documents in this dataset, 243 of which are labeled as positive (relevant). We follow the experimental procedure delineated above. The reviewer provided us with 43 ranked positive and 26 ranked negative features. There were five discrete groups, or sets, of ranked positive terms. The terms in the most positive group were thus five times as indicative of a relevant citation as those in the least positive group. The expert also provided three sets of ranked negative terms. We show results for up to 1,000 labels, at which point the performance of the classifiers asymptotes. The first significant observation is that both the CoFeature and CW-SVM Co-Testing based approaches dominate baseline methods until $\sim$ 600 queries, at which point Simple catches up. The second important observation is that the CW-SVM based method is able to exploit ranked features in early active learning rounds to outperform CoFeature, suggesting that there may be some benefit in exploiting rankings of features during AL, especially when there are very few labeled instances.

As formulated in Chapter 3, the CW-SVM requires specifying values for several parameters. For this particular experiment $\kappa = 0.1, c_1 = 0.75, c_2 = 0.1, c_3 = 0.2$. As each weight parameter was bounded between the range $-100 \leq w_i \leq 100$ to cover 7 rankings, we selected user defined values that balanced expected gap sizes with empirical error without significant parameter tuning due our limited data setting. With these settings, the CW-SVM method slightly outperforms CoFeature from 50-150 and 200-300 queries. Our results with other parameter settings show that if we set the parameters to bias the CW-SVM to heavily favor domain knowledge, then we see large gains during early rounds, but performance plateaus more slowly. If, on the other hand, we set the parameters to bias the CW-SVM toward minimizing empirical error then we observe a less pronounced early jump with a steadier performance increase. We thus conclude that the best use of rankings is to begin with a strong bias toward agreeing with

the ranking and decrease this importance as labeling proceeds. Intuitively this makes sense: as additional evidence accrues in the form of labeled instances, prior information becomes less important.

5.5 Conclusions

We have presented the dually supervised active learning paradigm, which looks to exploit labeled features during AL. This is a nascent area receiving an increasing amount of attention in the literature, as evidenced by our review of emerging work in this direction (Section 5.1). In Section 5.2 we discussed when such external information may improve model performance, particularly in the case of imbalanced scenarios. We presented a novel dually supervised active learning strategy that extends the Co-Testing framework to exploit labeled features in Section 5.3. We also incorporated our CW-SVM, presented in Chapter 3, to actively learn from ranked labeled features. We demonstrated that these strategies outperform baseline active learning methods in Section 5.4.

We have now presented several methodological contributions that look to improve machine learning in realistic application scenarios. In the next chapter we will now turn our focus to the practical application of the methods we have developed to our motivating task of citation screening for systematic reviews.

6

Toward Modernizing the Systematic Review Pipeline

In preceding chapters, we introduced methodological advancements that address deficiencies in current state-of-the-science machine learning techniques when applied to real-world tasks. We have used the task of citation screening as our motivating scenario throughout, but have thus far emphasized the broad applicability of the proposed methods, i.e., we have considered the task primarily from a machine learning, rather than a clinical research, vantage. In this chapter we focus more specifically on the practical task of semi-automating citation screening, the application of the developed technologies to this problem, and the implications of this for the systematic review process. Portions of this chapter have appeared in *Genetics in Medicine* (162), *BMC Bioinformatics* (168) and the *2011 Proceedings of the International Conference on Health Informatics* (164).

We first report results from a realistic prospective evaluation of our semi-automated approach when applied to the task of updating existing systematic reviews in Section 6.1. We demonstrate that machine learning can indeed substantially reduce workload, without sacrificing thoroughness. In Section 6.2, we then present our open-source, web-based software for citation screening, *abstrackr*, which provides a means of disseminating the machine learning methods that we have developed in this thesis. In our view the step of deploying machine learning technologies in order to actually make them useful to experts is too often overlooked by researchers.

6.1 Reducing the Workload Required to Update Systematic Reviews

A problem with systematic reviews is that their conclusions are sound with respect only to the evidence available at the time the review was conducted. Ideally, reviews would be updated each time new relevant evidence is published. But it is estimated that more than half of the systematic reviews in the Cochrane Library, a repository of high-quality systematic reviews, have not been updated for at least two years (68). Furthermore, a recent survey of organizations that produce and maintain systematic reviews suggests that at least half of existing reviews are already out of date, limiting their utility (67). The main reason for the staleness of reviews is the labor required to update them.

Here we demonstrate that the adoption of the proposed machine learning approach to semi-automating citation screening can eliminate a substantial amount of the work involved in updating reviews, thereby saving time and human resources, and ultimately increasing the likelihood that reviews are kept current. The active learning methods we have developed are not applicable to the task of updating systematic reviews due to the nature of the task. Because we are updating an existing review, we have at hand many examples of relevant and irrelevant papers; namely those citations screened for the original review. There is thus no need to acquire additional training data.

In earlier work (168) we demonstrated that a variant of active learning can reduce workload by half in the case of *de novo* reviews, without missing any relevant citations, i.e., any of the studies found relevant at level-2 screening (see Section 1.1) and thus ultimately included in the final review. This section demonstrates the practical benefits of applying the undersampling/bagging method we developed in Chapter 2 with respect to reducing the workload required to update systematic reviews.

In practice we have found that using committees of classifiers induced over different views of the data – the title, abstract and MeSH keywords[1] – further improves performance (168). To aggregate the predictions over these views,

[1]MeSH stands for Medical Subject Headers.

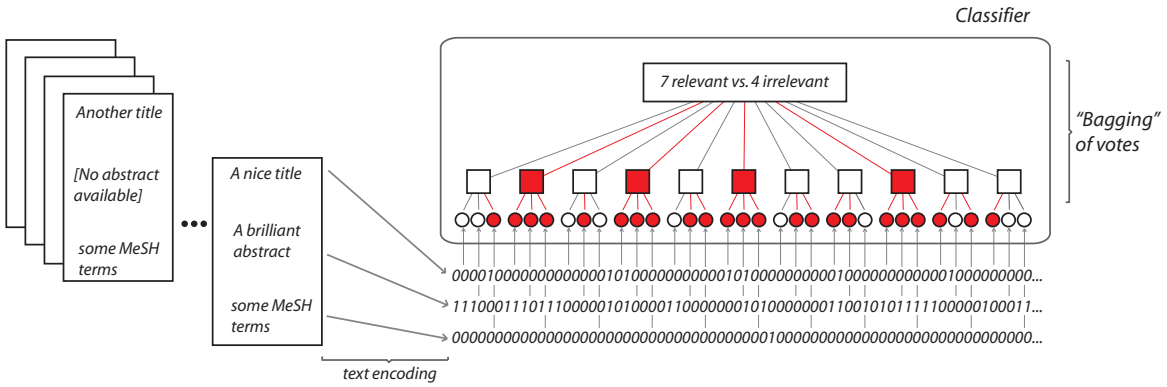


Figure 6.1: We encode the titles, abstracts, and MeSH terms components of citations using the standard "bag-of-words" representations. We used an ensemble of eleven base classifiers (squares) comprising three Support Vector Machines (SVMs, circles), one per encoded component. White circles and red circles stand for SVMs that classify their respective encoded components as relevant and irrelevant, respectively. If at least one of the SVMs suggests that the citation is relevant, the corresponding base classifier casts a relevant vote (white squares), otherwise it casts a vote for irrelevant (red squares). The overall disposition is given according to the majority vote of the ensemble of eleven base classifiers (here, relevant with seven versus four votes). The proportion of votes for the "winning" disposition is a proxy for the confidence of the classifier in its ultimate vote (here 7/11=0.64).

we take the simple approach of classifying citations as relevant iff *any* of the committee members deems it as such. In summary, we bag eleven ensemble classifiers, each induced over independently drawn balanced bootstrap samples from the training set comprising the citations screened for the original review.[1] This approach is described schematically by Figure 6.1; see its caption for more details.

6.1.1 Datasets

We used four systematic review datasets to validate our approach. Three synthesize genetic association studies, investigating Parkinson's disease (PDGene; `http://www.pdgene.org`(103)), Alzheimer's disease (AlzGene; `http://www.alzgene.org` (94)) and schizophrenia (SzGene; `http://www.szgene.org`(3)), respectively. These are summarized in Table 6.1. The fourth is the Tufts Cost-Effectiveness Analysis Registry (CEA Registry; `https://research.tufts-nemc.org/cear4`), which summarizes information from published cost-effectiveness analyses. Cru-

[1] We do not use eleven for any special reason; it is just a reasonable committee size that has worked well in the past. Initial explorations on previous datasets have suggested that beyond this point, adding additional members does not much alter performance.

	Training set (inception – 2009)		Update (validation) set (2010)	
Dataset	Size	Included (%)	Size	Included (%)
PDGene	20,216	556 (2.8)	561	104 (19)
AlzGene	42,833	1287 (3.0)	7298	65 (0.9)
SzGene	25,804	1410 (5.5)	5381	179 (3.3)
CEA Registry	5114	2287 (44.7)	1015	79 (7.8)

Table 6.1: Training and update (validation) sets in the four systematic reviews.

cially, none of these datasets were used during the development of the machine learning approaches that we have developed. Thus this evaluation truly is equivalent to a prospective application of the semi-automatic approach.

The protocol and methods for our four datasets are available on their respective websites. In contrast to typical systematic reviews, these address much broader questions, and are updated on a weekly or monthly basis. For example, the AlzGene review evaluates the strength of the association between Alzheimer's disease and genetic variations across the whole genome, whereas a typical systematic review would probably evaluate only a subset of such genetic variations, e.g., in the APOE gene. Note that attaining perfect (100 percent) sensitivity with semi-automated updating is much more difficult when all reported variations across thousands of genes are of interest rather than only APOE variations.

To simulate a prospective test of our semi-automated system, we segmented each of the datasets into a training set, comprising all citations published through 12/31/2009 and an update (validation) set, composed of citations published between 01/01/2010 and 12/31/2010. This is equivalent to a prospective evaluation of our semi-automated system throughout 2010.

For each dataset, we calculated the sensitivity and specificity of the classifiers on the update set. The reference standard was whether a citation was ultimately included in the systematic review or not during manual screening, i.e., whether it passed level-2 screening. We report the number of citations that reviewers would have needed to screen, had they been using the proposed semi-automated system to update reviews in 2010, versus the number of citations they actually screened. We assessed the variability of overall results by repeating all analyses twenty times using different random number seeds.[1] We arbitrarily considered

[1]While the training and test sets are fixed, recall that undersampling introduces randomness.

Dataset	TP	FN	Sensitivity (range)	TN	FP	Specificity (range)
PDGene	104	0	100 (100, 100)	5011	501	90.9 (90.0, 91.1)
AlzGene	65	0	100 (100, 100)	6743	490	93.2 (93.0, 93.2)
SzGene	179	0	100 (100, 100)	4664	538	89.7 (89.2, 89.7)
CEA Registry	78	1	98.7 (98.7, 98.7)	680	256	72.6 (72.1, 73.0)

Table 6.2: TP: True positives (citations deemed relevant by the classifier and included in the systematic review [upon full text review]); FN: false negatives (citations deemed irrelevant by the classifier but were included in the systematic review); FP: false positives (citations deemed relevant by the classifier but were not included in the systematic review); TN: true negatives (citations deemed irrelevant by the classifier and were not included in the systematic review).

the first run as the main analysis, and report minimum and maximum results from the other nineteen.

6.1.2 Results

In all three genetic topics the proposed semi-automated strategy correctly identified all citations that were included in the systematic reviews in 2010 (100% sensitivity), and considered relevant only approximately 10% of the papers that were excluded by the human experts (specificity of about 90 %). Had the semi-automated system been used in 2010, the human experts would have needed to screen only 605 (PDGene), 555 (AlzGene) and 717 (SzGene) titles and abstracts, compared to the 5616, 7298 and 5381 citations they manually screened for the three datasets (Table 3). This translates to reductions in labor of approximately 81, 92 and 87 percent, respectively.

In the case of the CEA Registry, the classifier missed only one eligible article (sensitivity about 99 percent), and incorrectly considered relevant approximately 28 percent of the papers that were excluded by human reviewers in 2010 (specificity around 73 percent). Relying on the semi-automated system throughout 2010, researchers would have needed to screen only 334 out of 1015 citations (a reduction in labor of approximately 67 percent). Upon re-review of the single false negative, human experts deemed that this citation might also have been missed by a novice human reviewer: only a single sentence in the abstract suggests that a cost-effectiveness (or cost-utility) analysis might have been performed, i.e., that it was indeed relevant.

All results were robust when we repeated the entire analysis an additional

nineteen times using different random number seeds (recall that undersampling introduces randomness into the induction process). No eligible papers were missed in the three genetic topics, and the same eligible paper was always missed in the CEA Registry. The specificity of the classifiers was nearly identical to the main analyses (Table 6.2).

We have shown that machine learning methodologies can indeed reduce the burden of updating of systematic reviews without sacrificing their comprehensiveness. Only a single citation out of the many dozens that were included in each topic's 2010 update would have been missed by the semi-automated method, and this was a borderline case.[1] This is directly comparable to the performance of individual human screeners: in empirical explorations human experts missed on average 8 percent of eligible citations (ranging from 0 to 24 percent) (61). To minimize the likelihood of overlooking eligible studies, current recommendations suggest using two independent screeners. Thus, computer assisted screening could replace full-manual screening for both screeners, replace one screener, or could be used in addition to both screeners to further increase the sensitivity of the overall process.

We have thus demonstrated via a realistic prospective empirical evaluation that machine learning can indeed be of practical use. This is welcome news, but if such technologies are to be adopted in practice then tools must be made available to the clinical researchers conducting the reviews. We next describe our work on *abstrackr*, an open-source, web-based annotation tool for citation screening that integrates our machine learning tools in a GUI-based tool for conducting systematic reviews.

6.2 Putting it all Together: the *abstrackr* System

The data deluge in clinical science has motivated the development of machine learning and data mining technologies to facilitate efficient biomedical research (40, 168, 184). Despite the obvious potential of such methods and the concomitant academic interest therein, however, adoption of machine learning tech-

[1] Moreover, the one relevant citation that was missed belonged to the review for which we had the smallest amount of training data.

niques by medical researchers has been relatively sluggish. One explanation for this is that while many machine learning methods have been proposed and retrospectively evaluated, they are rarely (if ever) actually made available to the practitioners whom they would benefit. In this section we describe the ongoing development of an end-to-end interactive machine learning system at the Tufts Evidence-based Practice Center (EPC). More specifically, we have developed *abstrackr*, an open-source, web-based tool for the task of citation screening for systematic reviews. This tool provides an interface to our machine learning methods. The *abstrackr* program thus provides a means of deploying the novel machine learning techniques described in this thesis.

abstrackr (accessible at `http://abstrackr.tuftscaes.org`; source code available via GitHub `https://github.com/bwallace/abstrackr-web`) is a collaborative (i.e., multiple reviewers can simultaneously screen citations for a review), web-based annotation tool for the citation screening task. It supports interactive learning protocols such as active learning and dual supervision, in addition to other forms of annotation, such as note-taking (see Figure 6.2). Ultimately, our goal in developing *abstrackr* has been to create a practical means of deploying the machine learning technologies that we have developed to researchers undertaking systematic reviews, i.e., screening citations. But because we have not yet conducted a large-scale empirical evaluation of our methods for semi-automating the citation screening process, *abstrackr* is currently primarily used as an annotation tool. (Reviewers will not trust the system to screen citations on its own without such a large-scale empirical validation).

Even without the machine learning components, *abstrackr* has been found useful by the Tufts Evidence based Practice Center (EPC), where it is currently being routinely used. Moreover, the active learning elements are already being regularly used to order citations with respect to their likelihood of being relevant, expediting the citation screening process. This tedious screening task was previously being conducted by printing out reams of abstracts to read one-by-one while keeping track of screening decisions – labels – in a spreadsheet (see Figure 1.2). As one might imagine, this was a messy and generally unenjoyable endeavor. *abstrackr* also provides a digital paper trail, and is helpful in tracking

and managing workload – i.e., assigning citations to reviewers. Because of this *abstrackr* has been found useful as a stand-alone annotation tool, independent of the machine learning components. We have thus had domain experts willing to use our software. This has been helpful, because they have provided rapid feedback regarding the software. More importantly, this has provided empirical data (some of the datasets in Table 2.2) and a platform on which to distribute the machine learning methods we develop.

A typical work-flow in *abstrackr* proceeds as follows. First, a literature search is conducted in the usual way, e.g., via PubMed. Once the set of potentially eligible citations is retrieved, it is imported into *abstrackr*. This starts a new review/project. The user who creates a review is designated as its lead. During the review creation process, project leads are asked a few questions about the project. In particular, they are asked in which order citations are to be prioritized for screening – here they are effectively specifying the active learning function to be used. For example, they may elect to screen citations in order of the likelihood that they are relevant, as predicted by the current model (we use the scoring function $\mathcal{Q}$ discussed in Chapter 4). The former is the default, but project leads may alternatively elect to simply screen the citations in a random order. Once the review is created, the lead can invite other reviewers to join the project.

Reviewers will spend the majority of their time interacting with the interface shown in Figure 6.2. In this interface, they are presented with a citation (title, abstract and MeSH keywords) and can designate it as 'relevant', 'borderline' or 'irrelevant'. Once one of these labels is assigned to the citation, the reviewer is immediately presented with a new citation to screen. The next citation selected by the system is a function of the active learning strategy $\mathcal{Q}$ that was selected for the corresponding review.

Terms and n-grams that the user has labeled are highlighted in a color indicating their polarity, i.e., whether (and to what degree) the highlighted term is indicative of 'relevance' or 'irrelevance'. Initial interactions with reviewers suggested that it is natural for them to provide two levels of granularity in either direction, i.e., a given term might be designated as 'highly' or 'weakly' indicative of relevance (irrelevance). Users can add additional labeled terms at the bottom

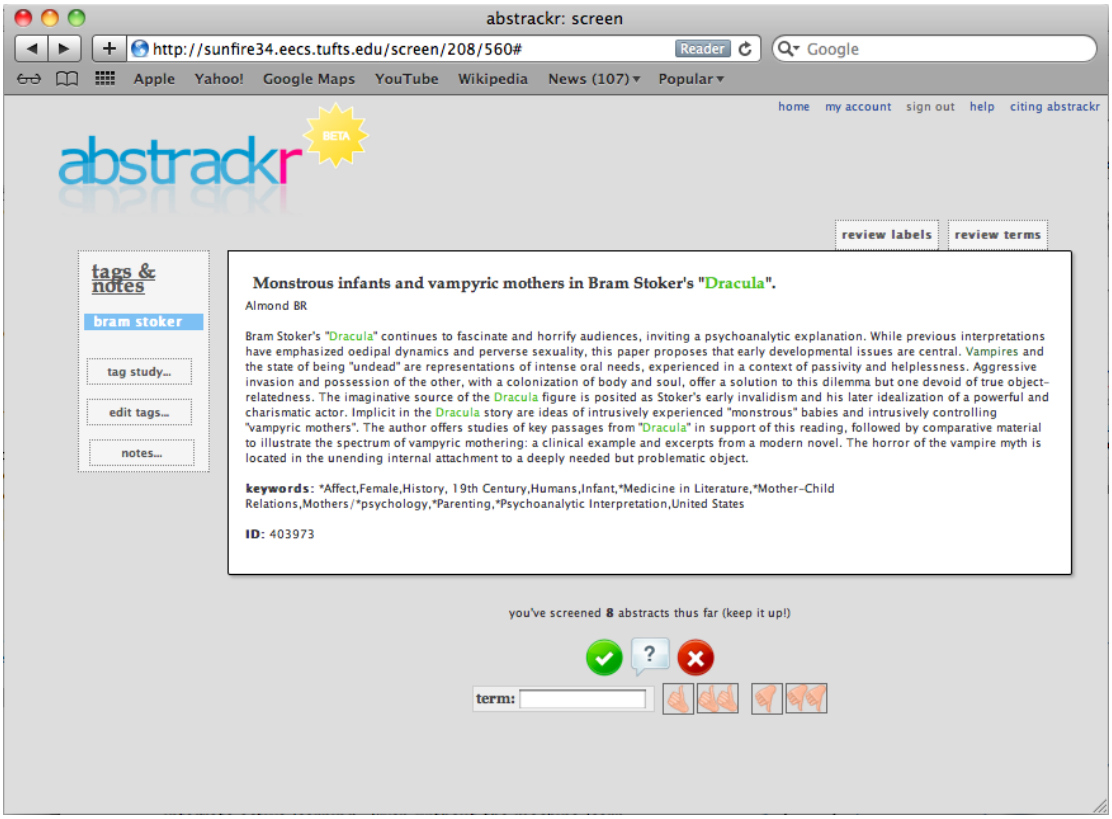


Figure 6.2: The main user interface of the *abstrackr* software. Terms that the expert has designated as indicative of relevance or irrelevance are highlighted (green for positive/relevant, red for negative/irrelevant). Users may enter additional terms into the text-box at the bottom of the screen, designating them as relevant (irrelevant) or strongly relevant (irrelevant) by clicking the single and double thumbs up (down) buttons, respectively. This 'thumb-level' encodes the rankings exploited by our CW-SVM (151); see Chapter 3. The labeled terms also inform the order in which the remaining abstracts will be shown to the reviewer, as described in Chapter 5. The reviewer can elect to accept (✓), designate as borderline/ambiguous (?), or reject (×) the current citation: these are the instance labels. Once they do so, the next citation (as ordered by the active learning ordering function) will immediately be retrieved and displayed to the user.

of the page; the thumb icons correspond to the aforementioned feature-labels. This interface enables dual supervision, discussed at length in Chapters 3 and 5. Experts can label both instances (citations) and features (words/n-grams). Both will ultimately be exploited by the CW-SVM (151) we formulated in Chapter 3. In addition to allowing users to impart labeled features, the *abstrackr* interface allows them to make other annotations regarding particular citations, including structured and general notes about studies and 'tags' that may be viewed as secondary labels. For example, users may tag all randomized control trials.

Figure 6.3 provides a schematic of the *abstrackr* system architecture. The numbered arrows in the figure indicate interactions and the general 'flow' of the system,

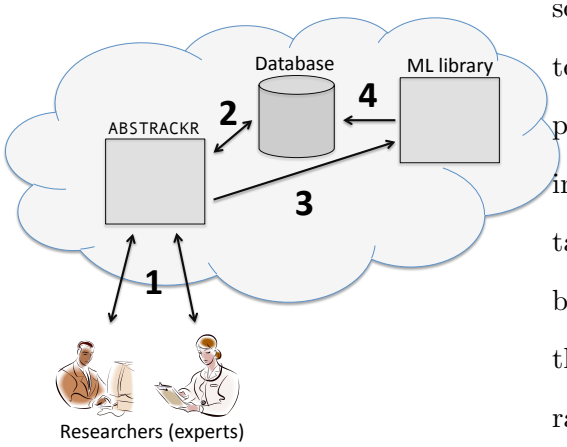


Figure 6.3: The abstrackr system architecture.

which we now describe. (**1**) Researchers undertaking the review interact directly with the web application via the interface depicted in Figure 6.2. (**2**) The next citation to be screened is selected based on a priority table stored in the database. This table contains ranked lists of citations for each review in the system; these citations are ranked according to the active learning function (e.g., uncertainty sampling) selected for the corresponding review. Re-training a model on all of the labeled data for a given review in order to re-calculate the active learning score for each instance in the unlabeled pool can incur a substantial computational cost; thus re-prioritizing the unlabeled citations each time a new citation is labeled can be quite slow. Any deployed active learning system must address this issue, or else it risks being unresponsive, thereby undercutting the aim of making better use of expert time. Our strategy is to perform this re-ranking asynchronously: (**3**) *abstrackr* periodically calls on the machine learning library (also local to the server) to (**4**) re-sort the citations for the current review. This asynchronous re-ranking means that the reviewer does not have to wait for the computer to decide which citation should be screened next; this is decided beforehand and immediately displayed to them.

In Chapter 4, we discussed the need to allocate labeling tasks in a way that makes the best possible use of the participating experts. The *abstrackr* system roughly follows the Multiple Expert Active Learning (MEAL) algorithm proposed in Section 4.3. This method requires a ranking of the participants with respect to expertise. That is, we need to know which of the participating

screeners are likely to provide high-quality labels and how pricey these labels will be. We assume that expertise correlates with cost, i.e., that less experienced (cheaper) reviewers will tend to provide lower-quality labels compared to more experience (expensive) reviewers. As a proxy for this information, we ask users how many systematic reviews they have previously participated in when they register for an account on *abstrackr*. When an inexperienced reviewer labels a citation with the '?' button, indicating that he or she is insufficiently confident as to whether it ought to be included or not, this citation is then re-assigned to a more experienced reviewer. Citations labeled as '?' are called 'maybes' within the system and can be reviewed at any time by the project lead, who must eventually make a screening decision. We have not yet integrated the predicted annotation time model described in Section 4.4 during active learning, but plan on doing so in the near future.

abstrackr has been used to facilitate screening in well over fifty systematic reviews. Once we have performed a large-scale validation of our machine learning approach to semi-automating screening, this functionality will be integrated into *abstrackr*. Specifically, the system will automatically screen out irrelevant citations, thus reducing workload.

We have built *abstrackr* to accommodate the interactive machine learning technologies introduced in this thesis. As already mentioned, *abstrackr* prioritizes the screening of citations with respect to the user-selected active learning criteria. The tool also makes nightly predictions regarding the likelihood that the remaining unscreened citations for a given review are relevant. These are estimated using the methodology developed in 2.3. Figure 6.4 shows the histogram displayed by *abstrackr* to summarize these estimates. At present, this is largely an exploratory tool that helps project leads estimate the expected workload remaining, based on how many citations are likely to be included in the review. As the machine learning technologies are more widely accepted by the systematic review community, we envision project leads using these probabilities to decide when to allow the system to automatically complete the screening.

Indeed, *abstrackr* has been used in two prospective cases already. However, because our large-scale validation remains to be performed, in these cases a

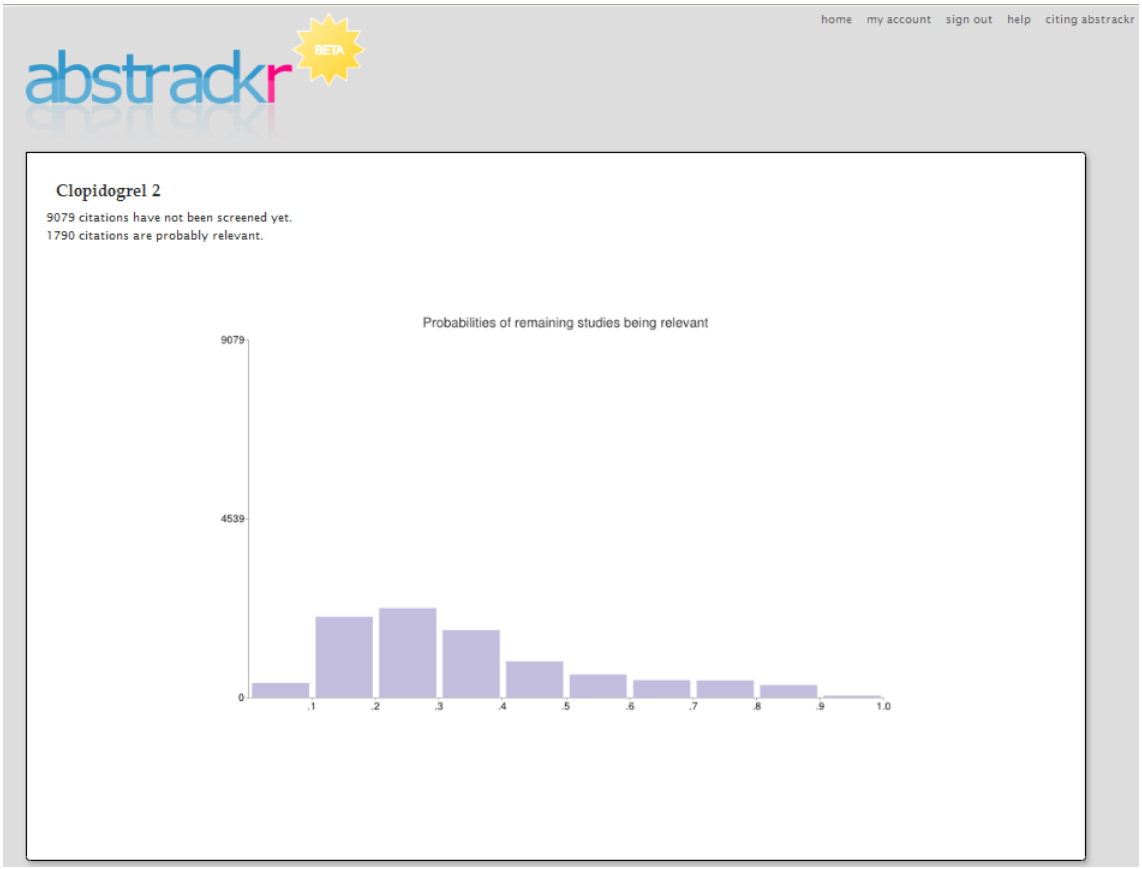


Figure 6.4: The *abstrackr* software displays a histogram of the predicted probabilities that remaining citations are relevant. These probabilities are estimated as described in 2.3. Here probabilities are shown for the 9,079 studies that remain in an ongoing systematic review regarding Clopidogrel being carried out by researchers at the Tufts EPC within the *abstrackr* software. As expected (given that relevant citations are rare) current probability estimates suggest that 1790 of the remaining 9079 citations are relevant ($\sim$20%). This can be seen by eyeballing the mass to the left of .5, which accounts for the majority of the remaining citations. Because we are relying on the method we developed and verified specifically for the task of predicting good class probabilities in imbalanced scenarios, we can be confident that those citations receiving low probability scores are indeed likely to be irrelevant.

trained assistant (not a physician) screened all of the citations that the algorithm excluded to double-check the classifier's decisions. When uncertain about a particular citation, the assistant deferred to the project lead (a physician; i.e., a more experienced reviewer). In both cases, *abstrackr* correctly included all relevant citations, that is, it never designated a relevant citation as being irrelevant. More specifically, we performed prospective classification for two reviews being conducted within *abstrackr*: one concerning treatments for sleep apnea, the other investigating self-measured blood pressure. In the former, 14,368 citations were retrieved via the initial query and had to be screened; in the latter 9,550 citations were retrieved. Using the *abstrackr* system, reviewers screened these citations interactively, in decreasing order of their likelihood of being relevant, as predicted by the machine learning model. We continued this process until the model no longer classified any of the remaining unlabeled citations as relevant. At this point, the remaining abstracts were screened by the assistant. To mitigate the possibility of false negatives on her part, the assistant was instructed to err on the side of inclusion, i.e., to mark for review by a more experienced expert any citations that were borderline or about which she was uncertain.

In the case of sleep apnea, 8,358 of the 14,368 (∼60%) of the citations were screened before the model predicted that the remaining 6,010 were irrelevant. The assistant marked for review 126 of these, all of which were subsequently excluded. For self-measured blood pressure, the model predicted that the remaining citations were irrelevant once 5,632 (again about 60%) were screened. At this point, the remaining 3,918 were screened by the assistant, who flagged 48 of these as being possibly relevant. Again, all 48 were subsequently rejected by the project lead.

In summary: on both reviews for which the classification component of the *abstrackr* system has been deployed prospectively, it reduced workload (the number of citations that needed to be manually screened) by about 40% without wrongly excluding any relevant reviews, i.e., the sensitivity of the classifier was 100%. This was verified by an assistant double-checking (screening) the citations that the system rejected. (Note that the assistant was explicitly instructed to err on the side of sensitivity). Once we have conducted our large-scale validation

on many real-world systematic review datasets, this latter step of manually verifying the classifier's decisions will no longer be required, assuming our method continues to replicate this caliber of performance. The results of this empirical evaluation should generalize because we are curating a diverse set of more than thirty systematic reviews.

6.3 Conclusions

In this chapter we have presented practical results regarding the application of the machine learning methodologies developed in this thesis to the task of citation screening for systematic reviews. In Section 6.1, we presented results from a realistic prospective evaluation of applying the semi-automated approach to update existing systematic reviews. We demonstrated that this can reduce workload substantially – by up to 90%, in some cases – without missing relevant articles. In Section 6.2 we described the *abstrackr* tool, which facilitates citation screening and implements the machine learning technologies developed in this thesis.

We are presently curating a large set of systematic review datasets in order to perform a large-scale verification of the proposed technologies for new reviews, in this future evaluation we will also exploit active learning and dual supervision for reviews for which labeled terms are available. This large-scale evaluation is critical because systematic reviewers need to trust that the system will not wrongly exclude relevant literature. By curating a large (more than thirty) set of diverse systematic reviews, and – hopefully – demonstrating that system consistently reduces labor without missing eligible studies, we will demonstrate this empirically.

7

Conclusions and Future Directions

There is a universe behind and before him. And the day is approaching when closing the last book on the last shelf on the far left: he will say to himself, "Now what?"

Jean-Paul Sartre, *Nausea*

Scientific disciplines are increasingly inundated with data. Domain experts in these subjects are relatively few, and their time and expertise is thus a scarce and valuable resource; making better use of domain expert time is crucial if we are to gain from the torrential amount of available information. Machine learning is an obvious candidate for reducing the labor required to squeeze useful knowledge from data. ML techniques can mine valuable facts from unstructured data, help experts find what they are looking for, and otherwise automatically or semi-automatically process information.

But if machine learning and data mining techniques are to be useful in practice, new methods must be developed to address the problems inherent to data mining in the real-world. This thesis has made several methodological contributions that aim to bring machine learning out of the lab and into practice, particularly in the context of domains that require substantial amounts of expensive human expertise. These methods have the over-arching aim of making better use of domain expert time, either by inducing better models in general, or

by exploiting novel interactive forms of supervision (i.e., via the active learning and dual supervision paradigms).

As a motivating application throughout this work, we have used systematic reviews. Systematic reviews look to rigorously identify and synthesize all of the literature relevant to a precisely formulated clinical question. A tedious step in the systematic review pipeline is retrieving from the body of biomedical literature the articles pertinent to the clinical question at hand. The number of published manuscripts that researchers must search through to find these articles is already enormous, and it continues to grow exponentially (81). But making sense of this data – that is, finding out what actually works in health-care – is arguably more important than ever, especially in light of impending health-care reform in the United States. Due in part to the growth of the literature and in part to increasingly rigorous standards, the amount of work necessary to produce and maintain these reviews is becoming unmanageable (18). Systematic reviews are but a single illustrative example; experts everywhere in health sciences and beyond are struggling with information overload.

Tasks in clinical informatics in particular pose challenges to existing machine learning technologies. Specifically, class imbalance is inherent to such tasks, and hinders the performance of 'off-the-shelf' learning algorithms, especially with respect to the rare class. Moreover, probability estimates from existing models are unreliable under imbalance. Another issue is that it is too costly, in terms of expert time, to label the amount of training data necessary to induce sufficiently good ML models. This problem may be mitigated by the emerging paradigms of active learning and dual supervision. Work in the former direction, however, has made several unrealistic assumptions that hinder its adoption in real-world tasks. Methods that take the latter approach, meanwhile, are only recently being proposed.

In this thesis we have developed novel machine learning and data mining methods that address these issues. The immediate aim has been to reduce the workload involved in conducting systematic reviews. But this is only an exemplary task; the approaches we have presented here have wider application to many real-world learning problems, i.e., those that require specialized ex-

pertise, exhibit class imbalance (and asymmetric costs) and for which limited human annotation resources are available. We have shown that the methods we have developed bring substantial improvements over previously existing machine learning approaches in terms of inducing better models with less human effort. We next summarize our contributions and sketch future research directions.

7.1 Thesis Contributions

The main data mining and machine learning contributions we have made are summarized as follows.

- **A better understanding of class imbalance and how to mitigate its effects on learning.** Imbalance is a property inherent to many interesting real-world learning tasks, including biomedical citation screening. Yet despite a wealth of research investigating supervised learning in imbalanced scenarios (77), there has been little theoretical understanding of the problem. Consequently, existing methods for handling imbalance are not well-motivated, and thus unreliable. In Chapter 2, we developed a theoretical framework for probabilistically analyzing imbalanced scenarios. We used this framework to motivate the strategy of bagging an ensemble of classifiers induced over balanced bootstrap training datasets (166). We demonstrated empirically when this strategy will work well, compared to alternative methods for handling imbalance. Furthermore, we considered the task of making class probability estimates in imbalanced scenarios, a problem that has previously received little attention. We demonstrated that in imbalanced scenarios, probability estimates are inherently biased, i.e., they tend to underestimate the probability that rare instances indeed belong to the minority class. We introduced the stratified Brier-score as a metric to quantify the class conditional performance of probability estimators, and we proposed undersampling and bagging as a means of mitigating this bias, thereby inducing better probability estimators. Finally, in Chapter 4, we proposed a novel means of evaluating the performance of

classification systems in imbalanced scenarios that elicits the relative costs of false positives (negatives) from the domain experts themselves (165).

- **New methods for learning under dual supervision.** Traditional supervised learning algorithms exploit only *instance labels*, i.e., each training example is associated with a single class label. However, domain experts may be able to provide more direct forms of supervision in the form of labeled features. A labeled feature is an attribute whose presence is indicative of class membership. For example, if one is inducing a model to discriminate positive from negative movie reviews, the presence of the word 'great' is likely to indicate membership in the former class, while the word 'terrible' suggests the latter. In Chapter 3 we developed a new method that extends the SVM model to exploit such information during classifier induction (151). Moreover, this model is flexible enough to facilitate learning from ranked labeled features: e.g., experts may signal that certain features are strongly indicative of one class, while others only weakly indicative.

- **Novel methods for real-world active learning.** In the canonical active learning scenario, it is assumed that there is a single, infallible 'oracle' who provides labels at a fixed cost. In reality, there are often multiple labelers of varying skill and cost participating in a task. Indeed, this is the case in biomedical citation screening; there are typically three to six reviewers on a given project, some of whom are experienced (expensive) and others who are relatively novice (cheap). This is a common scenario in specialized tasks. In Chapter 4 we proposed a novel active learning method that makes the best use of a given group of experts with varying cost and expertise, i.e., at each step in AL, we pick who is to do the labeling in addition to which instance is to be labeled (167). Furthermore, we proposed a novel method to predict the length of time – a proxy for cost – that it will take to label a particular citation, and incorporated this prediction into the instance selection process in AL (163). We demonstrated that this strategy makes better use of experts, i.e., produces better models with less annotator effort. Finally, in Chapter 5 we proposed a novel, co-testing based approach for

dually supervised active learning (165), and showed that exploiting dual supervision to guide the AL process can improve classifier performance, particularly in imbalanced scenarios.

- **Practical implications: reducing the workload in systematic reviews via data mining.** In Chapter 6 we discussed the practical implications of our work on the citation screening process. We demonstrated via a realistic empirical evaluation that *abstrackr* can reduce the number of citations that human experts must screen by eighty to ninety percent, representing a substantial reduction in labor, without missing any relevant citations. Furthermore, we have deployed the above technologies via an open-source, web-based system to facilitate abstract screening that we call *abstrackr*. The *abstrackr* tool, discussed at length in Section 6.2, implements multiple expert active learning and accommodates dual supervision. Most importantly, it allows us to actually deploy the technologies we have developed, thus turning theory into application.

7.2 Future Directions

We have made progress in developing machine learning methods that build better models with less human effort, and that address properties inherent to many real-world learning scenarios. We have shown that these methods are capable of substantially reducing workload in the case of citation screening for systematic reviews. Our contributions thus represent a promising step toward easing the burden on experts imposed by data overload. But as always, questions remain.

An immediate problem we are working on pertains to our dually supervised learning method, the CW-SVM (151). As mentioned, a drawback to this approach is the run-time necessary to solve the optimization problem as stated. Moreover, three C parameters must be estimated, rather than the two usually required for SVM. This increases the time required to perform grid search to estimate these. To make this approach more computationally feasible, we are therefore presently working on a Perceptron (136) formulation of the same intuitions that guide the CW-SVM. The aim is to achieve comparable performance

with a linear-time learning algorithm, thus making the application of dual supervision more practical.[1]

One avenue in the abstract screening problem that we did not explore in this thesis is the potential exploitation of the citation structure during (active) learning. We believe, for example, that it is likely that relevant studies tend to cite similar articles. It thus seems natural to consider this structure during model induction. This is referred to as collective classification (142). It would be interesting to explore alternative forms of supervision in collective classification. For example experts may communicate that they believe certain papers are 'hubs'. We are especially interested in the prospect of combining this with active learning approaches for collective classification, e.g., as proposed by Bilgic and Getoor (23).

An additional outstanding problem that we hope to soon address is the *concept drift* that occurs while updating systematic reviews. Concept drift (173) refers to scenarios in which the target concept changes over time. As an example, consider the task of updating existing systematic reviews (discussed in Section 6.1) in which we look to identify relevant articles published after the original review was conducted. In the case of medical treatments, one can imagine that the available treatments, technologies and other factors change over time. Thus the vocabulary in articles deemed relevant during the original review may not be the same as that found in newly published articles. This problem is closely related to *domain adaptation* (24, 50), in which one looks to adapt a classifier trained for one target concept to a related one. We have ignored this problem thus far, but it is likely that we could achieve better performance by accounting for this concept drift. We will first apply existing methods, and then potentially extend these to exploit specific properties of medical language.

Another problem we plan on tackling is the step after citation screening in the systematic review pipeline: data extraction. In this step, reviewers extract from the identified (relevant) studies the pieces of information that they wish to synthesize. This step is even more laborious than citation screening. We thus

[1]Note that this would still require selecting parameters via grid-search, but the induction algorithm would be linear-time rather than quadratic, which would greatly decrease running time.

plan to explore methods for automatic or semi-automatic extraction of clinically relevant information from free-text. Initial steps in this direction have already been made (89), but it is an extremely difficult task and a long way from being solved.

As previously mentioned, another immediate aim of ours is conducting a large-scale evaluation of the semi-automated citation screening approach. We are presently curating systematic review datasets, and aim to assemble at least thirty for the evaluation. This validation will allow systematic reviewers to trust the automated system's classification decisions. Our hope is to encourage wide-scale adoption of the tool, thereby saving large amounts of time and effort in aggregate.

We also hope to apply the technologies we have developed for the task of citation screening to other problems, but within health informatics and beyond. Indeed, we believe these methods are applicable to a wide range of real-world learning problems. Consider the task of legal document retrieval (117, 135), in which lawyers and other highly paid individuals must identify specific relevant information somewhere amidst torrents of documents. As in citation screening, the class of interest would be rare, experts would likely be highly skilled and possess domain knowledge, and active learning would like be a fruitful approach.

In the longer term, we hope to develop technologies to assist consumers of health information find trust-worthy, up-to-date clinical evidence relevant to their needs. Specifically, we hope to investigate machine learning technologies to make it easier for patients to discover reliable clinical literature regarding their condition(s). At present health information consumers are largely left to navigate the overwhelming volume of published literature by themselves.

Finally, in a related research direction, we hope to explore methods for automatically monitoring the quality and veracity of published clinical literature, as this task will increasingly become too onerous for humans. Operationally, a first step will be locating *adverse events* (i.e., undesirable health outcomes that occur during a clinical trial) in clinical texts. This would be immediately useful to clinicians looking for such information on a given drug. More interestingly, however, this extracted information could be cross-checked against the

corresponding entry on ClinicalTrials.gov (183), which is a database into which researchers are legally obliged to deposit details of clinical trials. Thus adverse events can sometimes be found in the ClinicalTrials.gov record but not in the published manuscript; there is more leeway in terms of what can be reported in the latter, and researchers are not always keen to report adverse events. This suggests the task of semi-automatically identifying underreporting of adverse events in clinical articles, perhaps by flagging manuscripts the model deems 'suspicious' and having a human review them.

Bibliography

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *ECML*, pages 39–50, 2004. 37, 38

[2] I. Allen and I. Olkin. Estimating time to conduct a meta-analysis from number of citations retrieved. *JAMA: The Journal of the American Medical Association*, 282(7):634, 1999. 27

[3] N. Allen, S. Bagade, M. McQueen, J. Ioannidis, F. Kavvoura, M. Khoury, and et al. Systematic meta-analyses and field synopsis of genetic association studies in schizophrenia: the szgene database. *Nature Genetics*, 40(7):827–834, 2008. 177

[4] V. Ambati, S. Vogel, and J. Carbonell. Active learning and crowd-sourcing for machine translation. In *Language Resources and Evaluation (LREC)*, 2010. 124

[5] D. Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987. 114

[6] D. Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988. 114

[7] D. Angluin and M. Kharitonov. When wonΓ t membership queries help? *Journal of Computer and System Sciences*, 50(2):336–355, 1995. 114

[8] S. Argamon-Engelson and I. Dagan. Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research*, 11:335–360, 1999. 117

[9] S. Arora, E. Nyberg, and C. Rosé. Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) Workshop on Active Learning for Natural Language Processing*, pages 18–26. Association for Computational Linguistics, 2009. 91, 92, 120, 146

[10] L. Atlas, D. Cohn, R. Ladner, M. El-Sharkawi, and R. Marks II. *Training connectionist networks with queries and selective sampling.* Morgan Kaufmann Publishers Inc., 1990. 114

[11] J. Attenberg, P. Melville, and F. Provost. A unified approach to active dual supervision for labeling features and examples. *Machine Learning and Knowledge Discovery in Databases*, pages 40–55, 2010. 162

[12] J. Attenberg, P. Melville, F. Provost, and M. Saar-Tsechansky. Selective data acquisition for machine learning. *Cost-Sensitive Machine Learning*, 2012. 162

[13] J. Attenberg and F. Provost. Why label when you can search?: alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 423–432. ACM, 2010. 127

[14] J. Attenberg and F. Provost. Inactive learning?: Difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, 12(2):36–41, 2011. 118

[15] J. Baldridge and A. Palmer. How well does active learning actually work?: time-based evaluation of cost-reduction strategies for language documentation. In *Empirical Methods on Natural Language Processing (EMNLP)*, pages 296–305. Association for Computational Linguistics, 2009. 120

[16] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *J. Mach. Learn. Res.*, 5:255–291, 2004. 164, 165

[17] M. Barza, T. A. Trikalinos, and J. Lau. STatistical considerations in meta-analysis. *Infect. Dis. Clin. North Am.*, 23:195–210, 2009. 25

[18] H. Bastian, P. Glasziou, and I. Chalmers. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9):e1000326, 2010. 27, 191

[19] E. Baum and K. Lang. Query learning can work poorly when a human oracle is used. In *International Joint Conference in Neural Netwroks*, 1992. 115

[20] B. Beigman Klebanov and E. Beigman. From annotator agreement to noise models. *Computational Linguistics*, 35(4):495–503, 2009. 132

[21] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 49–56. ACM, 2009. 118

[22] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88. ACM, 2007. 71

[23] M. Bilgic and L. Getoor. Effective label acquisition for collective classification. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 43–51. ACM, 2008. 195

[24] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 440, 2007. 195

[25] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998. 167

[26] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. 40, 43, 54, 78, 117

[27] N. Breslow, N. Day, et al. *Statistical methods in cancer research. vol. 1. the analysis of case- control studies.*, volume 1. Distributed for IARC by WHO, Geneva, Switzerland, 1980. 77

[28] G. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. 71, 74, 82

[29] C. Brodley and M. Friedl. Identifying and eliminating mislabeled training instances. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 799–805, 1996. 131

[30] C. Brodley, U. Rebbapragada, K. Small, and B. Wallace. Challenges and opportunities in applied machine learning. *Artificial Intelligence Magazine*, 2012. 24

[31] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006. 30

[32] P. Castaldi, M. Cho, M. Cohn, F. Langerman, S. Moran, N. Tarragona, H. Moukhachen, R. Venugopal, D. Hasimja, E. Kao, et al. The copd genetic association compendium: a comprehensive online database of copd genetic associations. *Human molecular genetics*, 19(3):526–534, 2010. 22, 66, 108

[33] P. Castaldi, M. Cho, M. Cohn, F. Langerman, S. Moran, N. Tarragona, H. Moukhachen, R. Venugopal, D. Hasimja, E. Kao, B. Wallace, C. Hersh, S. Bagade, L. Bertram, E. Silverman, and T. Trikalinos. The COPD genetic association compendium: a comprehensive online database of COPD genetic associations. *Human Molecular Genetics*, 2009. 138

[34] C. Chang and C. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011. 59

[35] N. Chawla, K. Bowyer, L. Hall, and W. K. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. 12, 39, 53, 56

[36] N. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004. 38, 44

[37] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer. Smoteboost: Improving prediction of the minority class in boosting. *Knowledge Discovery in Databases: PKDD 2003*, pages 107–119, 2003. 40

[38] M. Chung, E. M. Balk, S. Ip, G. Raman, W. W. Yu, T. A. Trikalinos, A. H. Lichtenstein, E. A. Yetley, and J. Lau. Reporting of systematic reviews of micronutrients and health: a critical appraisal. *Am. J. Clin. Nutr.*, 89:1099–1113, 2009. 22, 66

[39] D. Cieslak and N. Chawla. Analyzing pets on imbalanced datasets when training and testing class distributions differ. *Advances in Knowledge Discovery and Data Mining*, pages 519–526, 2008. 41, 75

[40] A. Cohen, W. Hersh, K. Peterson, and P.-Y. Yen. Reducing workload in systematic review preparation using automated citation classification. *JAMIA*, 13:206–219, 2006. 180

[41] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1):7–18, 2006. 37

[42] I. Cohen and M. Goldszmidt. Properties and benefits of calibrated classifiers. *Knowledge Discovery in Databases: PKDD 2004*, pages 125–136, 2004. 70

[43] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. 118

[44] C. Cole, G. Binney, P. Casey, J. Fiascone, J. Hagadorn, C. Kim, C. Wang, D. Devine, K. Miller, and J. Lau. Criteria for determining disability in infants and children: low birth weight. *Evidence Report/Technology Assessment No. 70. Prepared by New England Medical Center Evidence-based Practice Center under Contract No. 290-97-0019*, 2002. 26

[45] C. Cortes and V. Vapnik. Support-vector networks. 20(3):273–297, 1995. 34, 97, 98, 116

[46] C. Counsell. FOrmulating questions and locating primary studies for inclusion in systematic reviews. *Ann. Intern. Med.*, 127:380–387, Sep 1997. 25

[47] I. Dagan and S. Engelson. Committee-based sampling for training probabilistic classifiers. In *ICML*, pages 150–157, 1995. 114

[48] J. Dahl and L. Vandenberghe. CXVOPT - python software for convex optimization. 103

[49] S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008. 118

[50] H. Daumé. Frustratingly easy domain adaptation. In *Annual meeting-association for computational linguistics*, volume 45, page 256, 2007. 195

[51] M. DeGroot and S. Fienberg. The comparison and evaluation of forecasters. *The statistician*, pages 12–22, 1983. 71

[52] R. DerSimonian and N. Laird. Meta-analysis in clinical trials. *Controlled clinical trials*, 7(3):177–188, 1986. 25

[53] P. Donmez and J. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Conference on Information and Knowledge Management (CIKM)*, pages 619–628, 2008. 18, 120, 125, 126, 134, 143, 147

[54] P. Donmez, J. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2009. 125, 128

[55] M. Dredze and K. Crammer. Active learning with confidence. In *Proceedings of the 46th Annual Meeting of the Association for Computational*

Linguistics on Human Language Technologies: Short Papers, pages 233–236. Association for Computational Linguistics, 2008. 116

[56] M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning*, pages 264–271. ACM, 2008. 117

[57] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602, 2008. 88, 89, 94, 95, 107, 110, 159, 163

[58] G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *EMNLP*, pages 81–90. ACL Press, 2009. 159, 160

[59] C. Drummond and R. Holte. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Proceedings of the ICML Workshop on Learning from Imbalanced Datasets II*, 2003. 38, 54

[60] C. Drummond and R. Holte. Severe class imbalance: why better algorithms aren't the answer. In *ECML*, pages 539–546, 2005. 38, 121

[61] P. Edwards, M. Clarke, C. DiGuiseppi, S. Pratap, I. Roberts, and R. Wentz. Identification of randomized controlled trials in systematic reviews: accuracy and reliability of screening records. *Statistics in Medicine*, 21(11):1635–1640, 2002. 180

[62] B. Efron and R. Tibshirani. *An introduction to the bootstrap*. CRC Press, 1993. 55, 56

[63] C. Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 973–978, 2001. 71

[64] D. Firth. Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38, 1993. 77

[65] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999. 40, 117

[66] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2):133–168, 1997. 117, 167

[67] T. A. S. M. M. D. Garritty C, Tsertsvadze A. Updating systematic reviews: an international survey. *PLoS One*, 4(5), 2010. 176

[68] K. GG. No improvement - still less than half of the cochrane reviews are up to date. In *XIV Cochrane Colloquium*, 2006. 176

[69] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. Document classification through interactive supervision of document and term labels. *Knowledge Discovery in Databases: PKDD*, pages 185–196, 2004. 161

[70] H. Guo and H. Viktor. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM SIGKDD Explorations Newsletter*, 6(1):30–39, 2004. 40

[71] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou. On the class imbalance problem. In *ICNC*, pages 192–201, 2008. 33, 38, 42

[72] R. Haertel, K. Seppi, E. Ringger, and J. Carroll. Return on investment for active learning. In *NIPS Workshop on Cost Sensitive Learning*, 2009. 120, 145, 146

[73] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009. 33

[74] H. Han, W. Wang, and B. Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. *Advances in Intelligent Computing*, pages 878–887, 2005. 39

[75] D. Harrison, D. Rubinfeld, et al. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978. 28

[76] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The annals of statistics*, 26(2):451–471, 1998. 76

[77] H. He and E. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, pages 1263–1284, 2008. 33, 38, 42, 192

[78] J. He. *Analysis of Rare Categories*. Springer-Verlag New York Inc, 2011. 165

[79] S. Hido and H. Kashima. Roughly balanced bagging for imbalanced data. In *SDM*, pages 143–152, 2008. 40, 43, 55

[80] J. Hulse, T. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *ICML*, pages 935–942, 2007. 39, 40, 43, 54, 64, 71

[81] L. Hunter and K. B. Cohen. Biomedical language processing: what's beyond pubmed? *Mol Cell*, 21(5):589–594, 2006. 191

[82] N. Japkowicz. Learning from imbalanced data sets: a comparison of various stretegies. In *Proc. of the AAAI Workshop on Learning from Imbalanced Data Sets*, 2000. 38

[83] N. Japkowicz and S. Stephen. The class imbalance problem: a systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002. 33, 38, 42, 44, 66

[84] T. Joachims. Text categorization with support vector machines: learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142, 1998. 30, 170

[85] T. Joachims. *Learning to classify text using support vector machines: methods, theory and algorithms*, volume 186. Kluwer Academic Publishers Norwell, MA, USA:, 2002. 30

[86] D. Kahneman. *Thinking, fast and slow.* Farrar Straus & Giroux, 2011. 122

[87] P. Kang and S. Cho. EUS SVMs: ensemble of under-sampled svms for data imbalance problems. In *ICONIP*, pages 837–846, 2006. 40, 43

[88] G. King and L. Zeng. Logistic regression in rare events data. *Political analysis*, 9(2):137–163, 2001. 75, 76, 78

[89] S. Kiritchenko, B. de Bruijn, S. Carini, J. Martin, and I. Sim. Exact: automatic extraction of clinical trial characteristics from journal publications. *BMC medical informatics and decision making*, 10(1):56, 2010. 196

[90] A. Kosmopoulos, G. Paliouras, and I. Androutsopoulos. Adaptive spam filtering using only naive bayes text classifiers. In *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS)*. Citeseer, 2008. 116

[91] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: a review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006. 44

[92] J. Kruger and D. Dunning. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, 77(6):1121–1134, 1999. 135

[93] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *ICML*, pages 179–186, 1997. 37, 38, 47, 54

[94] B. L, M. MB, M. K, B. D, and T. RE. Systematic meta-analyses of alzheimer disease genetic association studies: the alzgene database. *Nature Genetics*, 40(7):17–23, 2007. 177

[95] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001. 159

[96] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002. 48

[97] J. Lau, E. Antman, J. Jimenez-Silva, B. Kupelnick, F. Mosteller, and T. Chalmers. Cumulative meta-analysis of therapeutic trials for myocardial infarction. *New England Journal of Medicine*, 327(4):248–254, 1992. 25

[98] D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–50. ACM, 1992. 30

[99] D. Lewis. Evaluating and optimizing autonomous text classification systems. In *SIGIR*, pages 246–254, 1995. 121

[100] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994. 114, 115

[101] P. Liang, M. Jordan, and D. Klein. Learning from measurements in exponential families. In *ICML*. ACM New York, NY, USA, 2009. 163

[102] R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932. 137

[103] C. Lill, J. Roehr, M. McQueen, F. Kavvoura, S. Bagade, B. Schjeide, and et al. Comprehensive research synopsis and systematic meta-analyses in parklinson's disease: The pdgene database. *(under review)*, 2012. 177

[104] H. Lin, C. Lin, and R. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276, 2007. 72

[105] W. Liu and S. Chawla. A quadratic mean based supervised learning model for managing data skewness. In *SDM*, 2011. 38

[106] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory under-sampling for class-imbalance learning. In *ICDM*, pages 965–969, 2006. 38, 40, 43, 52, 54

[107] T. Luo, K. Kramer, D. Goldgof, L. Hall, S. Samson, A. Remsen, and T. Hopkins. Active learning to recognize multiple types of plankton. In

Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 3, pages 478–481. IEEE, 2004. 114

[108] N. Mamitsuka. Query learning strategies using boosting and bagging. In *Machine learning: proceedings of the fifteenth international conference (ICML'98)*, page 1. Morgan Kaufmann Pub, 1998. 117

[109] G. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*, pages 870–878, 2008. 159

[110] C. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008. 30

[111] A. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002. 107

[112] A. McCallum, G. Mann, and G. Druck. Generalized expectation criteria. *Computer science technical note, University of Massachusetts, Amherst, MA*, 2007. 94, 95, 159

[113] A. Mccallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *International Conference on Machine Learning (ICML)*, pages 350–358, 1998. 117

[114] P. McCullagh and J. Nelder. *Generalized linear models*. Chapman and Hall, 1987. 76

[115] P. Melville, W. Gryc, and R. Lawrence. Incorporating background knowledge into text categorization for improved sentiment analysis. Technical report, Technical Report, IBM, 2008. 92, 93

[116] P. Melville, W. Gryc, and R. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1275–1284. ACM, 2009. 89, 106, 109, 110

[117] D. Merkl and E. Schweighofer. En route to data mining in legal text corpora: Clustering, neural computation, and international treaties. In *Database and Expert Systems Applications, 1997. Proceedings., Eighth International Workshop on*, pages 465–470. IEEE, 1997. 196

[118] M. Monard and G. Batista. Learning with skewed class distributions. *Advances in Logic, Artificial Intelligence and Robotics*, pages 173–180, 2002. 38

[119] I. Muslea, S. Minton, and C. Knoblock. Active learning with multiple views. *Journal Artificial Intelligence Research (JAIR)*, 27:203–233, 2006. 167

[120] A. Niculescu-Mizil and R. Caruana. Obtaining calibrated probabilities from boosting. In *Proc. 21st Conference on Uncertainty in Artificial Intelligence (UAI'05)*, pages 413–420, 2005. 41, 72, 75, 80, 83

[121] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632. ACM, 2005. 41, 70, 71, 72, 75, 80

[122] T. Osugi, D. Kun, and S. Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 330–337, 2005. 164, 165

[123] B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278, 2004. 91, 105, 110, 125, 131

[124] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008. 91

[125] E. Perrin, C. Cole, D. Frank, S. Glicken, N. Guerina, K. Petit, R. Sege, M. Volpe, P. Chew, C. MeFadden, D. Devine, K. Miller, and J. Lau. CRiteria for determining disability in infants and children: failure to thrive.

Evidence Report/Technology Assessment No. 72. Prepared by New England Medical Center Evidence-based Practice Center under Contract No. 290-97-0019, Mar 2003. 26

[126] C. Phua, D. Alahakoon, and V. Lee. Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59, 2004. 37

[127] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 71, 72, 126

[128] F. Provost. Machine learning from imbalanced data sets 101. In *Proc. of the AAAI Workshop on Learning from Imbalanced Data Sets*, 2000. 44, 71, 74

[129] S. Rabe-Hesketh and A. Skrondal. *Multilevel and longitudinal modeling using stata*. Stata Corp, 2008. 84

[130] H. Raghavan and J. Allan. An InterActive algorithm for asking and incorporating feature feedback into support vector machines. In *SIGIR*, pages 79–86, 2007. 161, 162

[131] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on features and instances. *J. Mach. Learn. Res.*, 7:1655–1686, 2006. 89, 161, 162, 170

[132] V. Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Valadez, L. Bogoni, and L. Moy. Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 889–896. ACM, 2009. 128

[133] U. Rebbapragada. *Strategic targeting of outliers for expert review*. PhD thesis, TUFTS UNIVERSITY, 2010. 127

[134] J. Reitsma, A. Glas, A. Rutjes, R. Scholten, P. Bossuyt, and A. Zwinderman. Bivariate analysis of sensitivity and specificity produces informative

summary measures in diagnostic reviews. *Journal of Clinical Epidemiology*, 58(10):982–990, 2005. 68

[135] H. Roitblat, A. Kershaw, and P. Oot. Document categorization in legal electronic discovery: computer classification vs. manual review. *Journal of the American Society for Information Science and Technology*, 61(1):70–80, 2010. 196

[136] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. 194

[137] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448, 2001. 118

[138] R. Schapire. A brief introduction to boosting. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 1401–1406, 1999. 40

[139] V. E. Schütze, H. and J. Pedersen. Performance thresholding in practical text classification. In *CIKM*, pages 662–671, 2006. 158, 164, 167

[140] B. Schölkopf, C. Burges, and A. Smola. *Advances in kernel methods: support vector learning*. The MIT press, 1999. 31

[141] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, and A. Napolitano. Building useful models from imbalanced data with sampling and boosting. In *Proc. 21st Annual FLAIRS Conference*, pages 306–311, 2008. 40

[142] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008. 195

[143] B. Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2010. 113

[144] B. Settles. Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1467–1478. ACL Press, 2011. 89, 94

[145] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079. Association for Computational Linguistics, 2008. 114, 118

[146] B. Settles, M. Craven, and L. Friedland. Active learning with real annotation costs. In *Proceedings of the Neural Information Processing Systems (NIPS) Workshop on Cost-Sensitive Learning*, pages 1069–1078. Citeseer, 2008. 120, 146, 156

[147] H. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992. 114, 117

[148] J. Shawe-Taylor and N. Cristianini. *Support vector machines*. Cambridge University Press, 2000. 30

[149] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Knowledge Discovery and Data mining (KDD)*, pages 614–622, 2008. 127

[150] S. Sivaraman and M. Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):267–276, 2010. 114

[151] K. Small, B. Wallace, C. Brodley, and T. Trikalinos. The constrained weight-space svm: learning with ranked features. In *the 28th International Conference on Machine Learning (ICML)*, 2011. 20, 35, 89, 183, 184, 193, 194

[152] S. Stumpf, V. Rajaram, L. Li, W. Wong, M. Burnett, T. Dietterich, E. Sullivan, and J. Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies*, 67(8):639–662, 2009. 161

[153] Q. Sun and G. DeJong. Explanation-augmented svm: an approach to incorporating domain knowledge into svm learning. In *Proceedings of the*

22nd international conference on Machine learning, pages 864–871. ACM, 2005. 92

[154] Y. Tang, S. Krasser, P. Judge, and Y. Zhang. Fast and effective spam sender detection with granular svm on highly imbalanced mail server behavior data. In *Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006. International Conference on*, pages 1–6. IEEE, 2006. 37

[155] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser. SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):281–288, 2009. 38, 39, 44

[156] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bassing and random subspace for support vector machines- based relevance feedback in information retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006. 38, 40, 43

[157] T. Terasawa, T. Dvorak, S. Ip, G. Raman, J. Lau, and T. A. Trikalinos. Charged Particle Radiation Therapy for Cancer: A Systematic Review. *Ann. Intern. Med.*, 2009. 22, 66, 97, 108, 136, 154

[158] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. volume 2, pages 45–66, 2002. 17, 35, 116, 164, 166

[159] V. Vapnik. An overview of statistical learning theory. *Neural Networks, IEEE Transactions on*, 10(5):988–999, 1999. 31

[160] Vickers and Elkin. Decision curve analysis: a novel method for evaluating prediction models. *Medical Decision Making*, 26:565–574, 2006. 122

[161] B. Wallace and I. Dahabreh. On probability estimates for imbalanced data. In *under review at UAI 2012*, 2012. 34

[162] B. Wallace, K. Small, C. Brodley, J. Lau, C. Schmid, L. Bertram, C. Lill, J. Cohen, and T. Trikalinos. Toward modernizing the systematic re-

view pipeline in genetics: efficient updating via data mining. *Genetics in Medicine*, 2012. 175

[163] B. Wallace, K. Small, C. Brodley, J. Lau, and T. Trikalinos. Modeling annotation time to reduce workload in comparative effectiveness reviews. In *Proceedings of the 1st ACM International Health Informatics Symposium (IHI)*, pages 28–35. ACM, 2010. 35, 113, 193

[164] B. Wallace, K. Small, C. Brodley, J. Lau, and T. Trikalinos. Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. In *Proceedings of the 2nd ACM International Health Informatics Symposium (IHI)*. ACM, 2012. 175

[165] B. Wallace, K. Small, C. Brodley, and T. Trikalinos. Active learning for biomedical citation screening. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 173–182. ACM, 2010. 35, 140, 145, 151, 153, 158, 159, 193, 194

[166] B. Wallace, K. Small, C. Brodley, and T. Trikalinos. Class imbalance, redux. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 754–763. IEEE, 2011. 34, 38, 75, 77, 192

[167] B. Wallace, K. Small, C. Brodley, and T. Trikalinos. Who should label what? Instance allocation in multiple expert active learning. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2011. 35, 113, 193

[168] B. C. Wallace, T. A. Trikalinos, J. Lau, C. E. Brodley, and C. H. Schmid. Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics*, 11, 2010. 142, 164, 175, 176, 180

[169] S. Walter. Small sample estimation of log odds ratios from logistic regression and fourfold tables. *Statistics in medicine*, 4(4):437–444, 1985. 77

[170] G. Weiss. Mining with rarity: a unifying framework. *Sigkdd Explorations*, 6(1):7–19, 2004. 38, 44, 71

[171] P. Wheeler, E. Balk, K. Bresnahan, B. Shephard, J. Lau, D. DeVine, M. Chung, and K. Miller. CRiteria for determining disability in infants and children: short stature. *Evidence Report/Technology Assessment No. 73. Prepared by New England Medical Center Evidence-based Practice Center under Contract No. 290-97-001*, 2003. 26

[172] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Proceedings of the 2009 Neural Information Processing Systems (NIPS) Conference*, pages 2035–2043, 2009. 128, 131

[173] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996. 195

[174] P. Wolfe. The simplex method for quadratic programming. *Econometrica: Journal of the Econometric Society*, pages 382–398, 1959. 33

[175] S. Wu, K. Lin, C. Chen, and M. Chen. Asymmetric support vector machines: low false-positive learning under the user tolerance. In *KDD*, pages 749–757. ACM, 2008. 38, 52

[176] C. Yang, J. Wang, J. Yang, and G. Yu. Imbalanced svm learning with margin compensation. *Advances in Neural Networks-ISNN 2008*, pages 636–644, 2008. 52

[177] W. Yang and X. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(4):597–604, 2006. 42

[178] A. Yessenalina, Y. Choi, and C. Cardie. Automatically generating annotator rationales to improve sentiment classification. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 336–341. Association for Computational Linguistics, 2010. 91

[179] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD*