Constructing Informative Prior Distributions from Domain Knowledge in Text Classification

Aynur Dayanik
DIMACS & Computer Science
Rutgers University
Piscataway, NJ
aynur@rutgers.edu

David D. Lewis
David D. Lewis Consulting
Chicago, IL
sigir06pk@daviddlewis.com

David Madigan
DIMACS & Department of Statistics
Rutgers University
Piscataway, NJ
dmadigan@rutgers.edu

Vladimir Menkov*
Aqsaqal Enterprises
Penticton, B.C. Canada
vmenkov@cs.indiana.edu

Alexander Genkin
DIMACS
Rutgers University
Piscataway, NJ
alexgenkin@iname.com

ABSTRACT

Supervised learning approaches to text classification are in practice often required to work with small and unsystematically collected training sets. The alternative is usually viewed as building classifiers by hand, using an expert's understanding of what features of the text are related to the class of interest. This is expensive, requires a degree of computational and linguistic sophistication, and makes it difficult to use combinations of weak predictors. We propose instead combining domain knowledge with training examples in a Bayesian framework. Domain knowledge is used to specify a prior distribution for parameters of a logistic regression model, and labeled training data is used to produce and find the mode of the posterior distribution. We show on three text categorization data sets that this approach can rescue what would otherwise be disastrously bad training situations, producing much more effective classifiers.

Categories and Subject Descriptors

1.2.6 [**Artificial Intelligence**]: Learning; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Experimentation

*Current affiliation Amazon.com

Keywords

knowledge-based, maximum entropy, MAP estimation

1. INTRODUCTION

Numerous studies show that effective text classifiers can be produced by supervised learning methods, including support vector machines (SVMs) [11, 14, 33], regularized logistic regression [9, 33], and other approaches [14, 27, 33]. Most of these studies have used thousands to 10's of thousands of randomly selected training examples. In operational text classification settings, however, small training sets are the rule, due to the expense and inconvenience of labeling, or skepticism that the effort will be adequately repaid.

To learn from a handful of training examples, one must either use a sufficiently limited model class or some additional regularization penalty to effectively constrain the models learnable with a small amount of data. Otherwise overfitting (learning accidental properties of the training data) will yield poor effectiveness on future data. On the other hand, strong general constraints on the models themselves limit the effectiveness of learnable classifiers.

This situation can be improved if one has advance knowledge of which classifiers are likely to be good for the class of interest. In text categorization, for instance, such knowledge might come from category descriptions meant for manual indexers, reference materials on the topics of interest, lists of features chosen by a domain expert, or many other sources.

Bayesian statistics provides a convenient framework for combining domain knowledge with training examples [3]. The approach produces a *posterior distribution* for the quantities of interest (e.g., regression coefficients). Per Bayes theorem, the posterior distribution is proportional to the product of a *prior distribution* and the *likelihood function*. In applications with large numbers of training examples, the likelihood dominates the prior. However, with small numbers of training examples, the prior is influential and priors that reflect appropriate knowledge can provide improved predictive performance. In what follows we apply this approach with logistic regression as our model and text classification

(in particular text categorization) as our application.

We begin by reviewing the use of logistic regression in text classification, and the Bayesian approach in particular (Section 2), then discuss previous approaches to integrating domain knowledge in text classification (Section 3). Section 4 presents our Bayesian approach, which is simpler and more flexible. Section 5 describes our experimental methods, while Section 6 presents our results. We find on three test categorization test collections, using three diverse sources of domain knowledge, that domain-specific priors can yield large effectiveness improvements.

2. BAYESIAN LOGISTIC REGRESSION

A logistic regression model is a linear model for the conditional log odds of a binary outcome, i.e.

$$p(y_i = +1|\boldsymbol{\beta}, \boldsymbol{x_i}) = \frac{\exp(\boldsymbol{\beta}^T \boldsymbol{x_i})}{1 + \exp(\boldsymbol{\beta}^T \boldsymbol{x_i})} = \frac{\exp(\sum_j \beta_j x_{i,j})}{1 + \exp(\sum_j \beta_j x_{i,j})}$$

where y_i encodes the class of example i (positive = $+1$, negative = -1) and $x_{i,j}$ is the value of feature j for example i (e.g. a within document term weight). We assume that j runs from 0 to d, the number of features, and that $x_{i,0} = 1.0$ for all i, i.e. the model has an intercept term.

A logistic regression training algorithm chooses a vector of model parameters $\boldsymbol{\beta}$ that optimizes some appropriate criterion function on a set of training examples for which y_i values are known. In the Bayesian MAP (maximum a posteriori) approach to logistic regression [9, 19], the criterion function is the sum of the log likelihood of the data and the log of the prior distribution of the regression coefficients,

$$l(\boldsymbol{\beta}) = (-\sum_{i=0}^{n} \ln(1 + \exp(-\boldsymbol{\beta}^T \boldsymbol{x_i} y_i)) + \ln p(\boldsymbol{\beta}),$$

where $p(\boldsymbol{\beta})$ is the prior probability for parameter vector $\boldsymbol{\beta}$ and we output a value $\hat{\boldsymbol{\beta}}$ (which may or may not be unique) that maximizes $l(\boldsymbol{\beta})$. The prior, $p()$, can be any probability distribution over real-valued vectors. MAP estimation is neither necessarily superior or inferior to other Bayesian approaches [28].

Logistic regression [7, 15, 19, 26, 33, 32] and, to a lesser degree, the similar probit regression [4], has been widely used in text classification. Regularization to avoid overfitting has been based on feature selection, early stopping of the fitting process, and/or a quadratic penalty on the size of regression coefficients. The last of these, sometimes called *ridge logistic regression*, can be interpreted as MAP estimation where $p(\boldsymbol{\beta})$ is a product of univariate Gaussians with mean 0 and a shared variance [19]. Recently, Genkin et al. [9] showed MAP estimation with a product of univariate Laplace priors, i.e. a *lasso* [29] version of logistic regression, was effective for text categorization.

3. PRIOR WORK

Feature extraction is one use of domain knowledge (most famously in spam filtering [18]). Creating better features is good, but one would like to guide the learner to use them. Domain knowledge can also be used to choose which features to use (feature selection). An old example is stopwords [24, 30], often deleted in content-based text classification, but specifically included in authorship attribution. Another is relevance feedback, where words from the user query are usually required to appear in the learned model [2, 24]. The downside of feature selection is that it cannot reduce the impact of a term without discarding it entirely.

Relevance feedback may also use textual queries as artificial positive examples to supplement labeled training data. However, a query (or in general a domain-informative text) may have different length, non-domain vocabulary, and non-textual features than a training document, which poses risks for learning. Finally, some relevance feedback algorithms (e.g. Rocchio [2]) use a query to set initial values of some or all classifier parameters, which are then updated by training data. This is a more flexible approach, but past algorithms have not dealt with words that are negative predictors, strong predictors of uncertain polarity, or varying degrees of confidence in predictors.

Several recent papers have modified learning approaches—naive Bayes [13, 16], logistic regression (fit with a boosting-style algorithm) [25], and SVMs [31]—to use domain knowledge in text categorization. All modify the base learning algorithm, and require users to convert knowledge about words into weighted training examples. Several heuristics are suggested for this weighting, but implicitly assume a substantial number of task documents (at least unlabeled ones) are available. A recent study [21] that upweights human-selected terms in SVM learning (by altering document vectors) is similar in spirit to our work, though in an active learning context.

Closely related to using domain knowledge is mixing training data from different sources in supervised learning (domain adaptation). Gabrilovich and Markovitch use a combination of feature generation, feature selection, and domain adaptation from a large web directory to improve classification of diverse documents [8]. Chelba and Acero [5] use out-of-task labeled examples in logistic regression training of a text capitalizer, and use the resulting MAP estimate as the mode vector of a Bayesian prior for training with in-task examples. Our work has similarities to Chelba and Acero's, as well as to non-textual uses of Bayesian priors to incorporate knowledge ([17], and citations therein).

4. USING DOMAIN KNOWLEDGE

Given the wide use in text classification of Gaussian priors, and the recent success of Laplace priors, we take these as our starting point. The univariate Gaussian and Laplace distributions each have two parameters, so a product of such distributions for d features and an intercept gives $2d + 2$ hyperparameters. The Gaussian parameters are the mean μ_j and the variance σ_j^2. The Laplace parameters are the mean μ_j, and the scale parameter λ_j, corresponding to a variance of $2/\lambda_j^2$. For both distributions the mean μ_j is also the mode, and in this paper we will refer to the mode and variance as the hyperparameters for both the Gaussian and Laplace distributions. The mode specifies the most likely value of β_j, while the variance specifies how confident we are that β_j is near the mode.

As for domain knowledge, our interest is in a wide range of possible clues to which words are good predictors for a class. Focused lists of words generated specifically for classification are of interest, but so are reference materials, such as encyclopedia entries, that provide noisier evidence. We refer to all these sources as "domain knowledge texts," and assume for simplicity there is exactly one domain knowledge text for each class (more can easily be used). We call a set of

such texts a "domain knowledge corpus." For a given class, we distinguish between two sets of words. Knowledge words (KWs) are all those that occur in the domain knowledge text for the class of interest. Other words (OWs) are all words that occur in the training documents for a particular run, but are not KWs. Table 1 summarizes the methods discussed in this section.

4.1 Baselines

Text classification research using regularized logistic regression has usually set all prior modes to 0, and all prior variances to a common value (or do the equivalent non-Bayesian regularization). Some papers explore several values for the prior variance [15, 33], others use a single value but do not say how it was chosen [19, 32], and others choose the variance by cross-validation on the training set [9]. We used cross-validation (Section 5.1.1) to choose a common prior variance for OWs. In our "No DK" baseline (Table 1), all words are OWs.

Another simple baseline is to create X copies of the prior knowledge text for a class and add these copies to the training data as additional positive examples ("DK examples" in Table 1), as in some relevance feedback approaches. We applied the same tokenization and term weighting (Section 5.3) to these artificial documents as to the usual training documents. We tested a range of values for X, but include results only for the best value, $X = 5$.

4.2 Priors From Domain Knowledge

Our four methods for using domain knowledge to specify class-specific hyperparameters begin by giving OWs a prior with $\mu_j = 0$ and a common variance σ^2 chosen by cross-validation. KWs are then given more ability to affect classification by assigning them a larger prior mode or variance than OWs. All four methods use a heuristic constant C_{DKRW}, the "domain knowledge relative weight", to control how much more influence KWs have. This constant can be set manually or, as in our experiments, chosen by cross-validation on the training set (Section 5.1.1).

Two of our methods look not just at the domain knowledge text for the target class, but at the texts for other classes, in order to determine how significant to the target class each word in its domain knowledge text is. As a heuristic measure of significance, we use TFIDF weighting (Section 5.3) within the domain knowledge corpus:

$$\text{significance}(t, Q) = \text{logtf}(t, d) \times \text{idf}(t), \qquad (1)$$

where

- d is the domain knowledge text for class Q,
- $\text{logtf}(t, d) = 0$ if term t does not occur in text d, or $1 + log_e(\text{tf}(t, d))$ if it does, where $\text{tf}(t, d)$ is the number of occurrences of t in d,
- $\text{idf}(t) = log_e((N_K + 1)/(\text{df}(t) + 1))$, where N_K is the total number of domain knowledge texts used to compute IDF weights, and $\text{df}(t)$ is the number of those documents that contain term t.

We now describe the methods.

4.2.1 Variance-Setting Methods

One view is that KWs will have more influence (i.e. parameter values farther from 0) than typical OWs in a good logistic regression model for the class, but could be positive or negative predictors. That suggests the prior on a KW should usually have a larger variance than the prior on an OW. Methods **Var** and **Var/TFIDF** (Table 1) make the prior variances for KWs a multiple of the variance for OWs. This multiple is the same for all KWs in Method Var:

$$\sigma_j^2 = C_{\text{DKRW}} * \sigma^2,$$

but is proportional to our heuristic measure of term significance (Equation 1) in Method Var/TFIDF:

$$\sigma_j^2 = C_{\text{DKRW}} \times \text{significance}(t_j, Q) \times \sigma^2.$$

Both methods use a prior mode of 0 for both OWs and KWs.

4.2.2 Mode-Setting Methods

Another view of a domain knowledge text is that it contains words which are mostly positive predictors of class membership, i.e. that KWs will tend to have parameter values greater than 0 in a good logistic regression model. Along these lines, Methods **Mode** and **Mode/TFIDF** make the prior mode for a KW greater than 0, in contrast to the mode of 0 used for OWs. Method Mode gives the prior for every KW the same mode:

$$\mu_j = C_{\text{DKRW}},$$

while Method Mode/TFIDF makes the prior modes proportional to term significance:

$$\mu_j = C_{\text{DKRW}} \times \text{significance}(t_j, Q).$$

Both methods use a common variance chosen by cross-validation for both OWs and KWs.

While mode-setting may seem more natural than variance-setting, it carries more risks. If a term does not occur in the training data, then the MAP estimate for the corresponding parameter is identically the prior mode. With nonzero prior modes and a tiny training set, we may be hardwire many untested parameter choices into the final classifier.

5. EXPERIMENTAL METHODS

In this section, we describe our experimental approach to studying the use of domain knowledge in logistic regression.

5.1 Software and Algorithms

As discussed in Section 3, our interest was in domain knowledge techniques that can be used with existing supervised learning algorithms. Here we discuss the particular implementations used in our experiments.

5.1.1 Logistic Regression

We trained and applied all logistic regression models using Version 2.04 of the BBR (Bayesian Binary Regression) package [9][1]. BBR supports Gaussian and Laplace priors with user-specified modes and variances. With Methods No DK and DK Examples we used prior modes of 0 and chose a common prior variance, σ^2, from this set of possibilities:

$$0.25, 1, 2.25, 4, 6.25, 9, 12.25, 16, 20.25, 25, 30.25, 36, 42.25, 49, 56.25, 64, 100, 10000, 1000000, 100000000.$$

The BBR fitting algorithm chose the prior variance that maximized the cross-validated posterior predictive log-likelihood for each training set.

[1] http://www.stat.rutgers.edu/∼madigan/BBR/

Method	Description of the method
No DK (baseline)	[KWs] - none [OWs, intercept] - mode: 0, variance σ^2 chosen by cross-validation
DK examples	Like No DK, but treat the domain knowledge text for the class as X positive examples
Var	[KWs] - mode: 0, variance: $\sigma_j^2 = C_{\text{DKRW}} \times \sigma^2$, ($C_{\text{DKRW}}$, σ^2) pair chosen by cross-validation [OWs, intercept] - mode: 0, variance: σ^2
Var/TFIDF	[KWs] - mode: 0, variance: $\sigma_j^2 = C_{\text{DKRW}} \times \text{significance}(t_j, Q) \times \sigma^2$ for term t_j and class Q, and (C_{DKRW}, σ^2) pair chosen by cross-validation [OWs, intercept] - mode: 0, variance: σ^2
Mode	[KWs] - mode: $\mu_j = C_{\text{DKRW}}$, variance: σ^2, (C_{DKRW}, σ^2) pair chosen by cross-validation [OWs, intercept] - mode: 0, variance: σ^2
Mode/TFIDF	[KWs] - mode: $\mu_j = C_{\text{DKRW}} \times \text{significance}(t_j, Q)$, variance: σ^2 for term t_j and class Q, and (C_{DKRW}, σ^2) pair chosen by cross-validation [OWs,intercept] - mode: 0, variance: σ^2

Table 1: Summary of tested methods for incorporating domain knowledge into learning. C_{DKRW} is a constant specifying the relative weight given domain knowledge.

For methods using class-specific priors, we used cross-validation external to BBR to choose a pair (C_{DKRW}, σ^2) from the cross product of a set of values for C_{DKRW} and the above set of values for σ^2. For Methods Var and Var/TFIDF, the C_{DKRW} values tried were 2, 5, 10, 20, 50, 100, and 10000. For Methods Mode and Mode/TFIDF, the C_{DKRW} values were 0.5, 1, 2, 3, 4, 5, 10, 20, 50, 100, and 10000. The pair was again chosen to maximize cross-validated posterior predictive log-likelihood on the training set.

5.1.2 Support Vector Machines

As a baseline to ensure that logistic regression was producing reasonable classifiers without domain knowledge, we trained support vector machine (SVM) classifiers on all training sets. SVMs are one of the most robust and effective approaches to text categorization [11, 12, 14, 27]. In our experiments, we used Version 5.0 of *SVM_Light* software [11, 12] [2]. All options were kept at their default values. Keeping the $-c$ option at its default meant that *SVM_Light* used the default choice ($C = 1.0$ for our cosine normalized examples) of the regularization parameter C. We also generated results with the regularization parameter chosen by cross-validation, but these were inferior and are not included here.

5.2 Datasets

Our text classification experiments used three public text categorization datasets for which publicly available domain knowledge texts was available. We chose, as our binary classification tasks, categories with a moderate to large number of positive examples. This enabled experimentation with different training set sizes.

5.2.1 Bio Articles

This collection of full text biomedical articles was used in the TREC 2004 genomics track categorization experiments [10].[3] The genomics track itself featured a few, atypical categorization tasks. However, because all the articles are indexed in the National Library of Medicine's MEDLINE system, they have corresponding MEDLINE records with manually assigned MeSH (Medical Subject Headings) terms. We posed as our text classification tasks predicting the presence or absence of selected MeSH headings.

Documents. We split the Bio Articles documents into three 8-month segments. We used the first segment for the training and the last segment for testing. The middle segment was reserved for future purposes and was not used in the experiments reported here. Training sets of various sizes were drawn from the training population of 3,742 articles (period: 2002-01-01 to 2002-08-31), and classifiers were evaluated on the test set of 4,175 articles (period: 2003-05-01 to 2003-12-31).

Categories. We wanted a set of categories that were closely related to each other (to test the ability of domain knowledge to support fine distinctions) and somewhat frequent on the particular biomedical journal articles we had available. MeSH organizes its headings into multiple tree structures, and we choose the A11 subtree (MeSH descriptor: "Cells") to work with. This subtree contains 310 distinct headings, and we chose to work with the 32 that were assigned to 100 or more of our documents. Note that when deciding whether a MeSH heading was assigned to a document, we stripped all subheadings from the category label.

Prior Knowledge. Each MeSH heading has a detailed entry provided as an aid to both NLM manual indexers and users of MEDLINE. Figure 1 shows a portion of one such entry. We used as our domain knowledge text for a category all words from the *MeSH Heading*, *Scope notes*, *Entry terms*, *See Also*, and *Previous Indexings* fields. Entries were taken from the 2005 MeSH keyword hierarchy [1], downloaded in November 2004.

5.2.2 ModApte Top 10

Our second dataset was the ModApte subset of the Reuters-21578 test collection of newswire articles [14].[4]

Documents. The ModApte subset contains 9603 and 3299 Reuters news articles in the training set and test set, respectively.

Categories. Following Wu and Srihari [31] (see below) we used the 10 "Topic" categories with the largest number of positive training examples.

Prior Knowledge. In their experiments on incorporating prior knowledge into SVMs, Wu and Srihari [31] manually specified short lists of high value terms for the top 10

[2] http://svmlight.joachims.org/

[3] http://trec.nist.gov/data/t13_genomics.html

[4] http://www.daviddlewis.com/resources/testcollections/reuters21578

MeSH Heading	Neurons
Tree Number	A08.663
Tree Number	A11.671
Annotation	do not use as a substitute or synonym for BRAIN / cytol
Scope Note	The basic cellular units of nervous tissue. Each neuron consists of a body, an axon, and dendrites. Their purpose is to receive, conduct, and transmit impulses in the NERVOUS SYSTEM.
Entry Term	Nerve Cells
See Also	Neural Conduction
...	...
Unique ID	D009474

Figure 1: A portion of MeSH entry for the MeSH heading "Neurons".

Class	Prior Knowledge
earn	cents cts net profit quarter qtr revenue rev share shr
acq	acquire acquisition company merger stake
money-fx	bank currency dollar money
grain	agriculture corn crop grain wheat usda
crude	barrel crude oil opec petroleum
trade	deficit import surplus tariff trade
interest	bank money lend rate
wheat	wheat
ship	port ship tanker vessel warship
corn	corn

Figure 2: Keywords used as prior knowledge for the ModApte Top 10 collection [31].

Topic categories. We used those lists (Figure 2) as our domain knowledge texts. Note that due to the small number of these texts and their highly focused nature, IDF weights within the domain knowledge corpus had almost no impact, so methods Var/TFIDF and Mode/TFIDF behaved almost identically to methods Var and Mode.

5.2.3 RCV1 A-B Regions

The third dataset was drawn from RCV1-v2, a test categorization test collection of 804,414 newswire articles [14]. [5]

Documents. For efficiency reasons, we did not use the full set of 804,414 documents. Our test set was the 120,076 documents dated 20-December-1996 to 19-February-1997. For a large training set, we used the LYRL2004 ([14]) training set of 23,149 documents from 20-August-1996 to 31-August-1996. Small training sets were drawn from a training population of 264,569 documents (20-August-1996 to 19-December-1996). The remaining documents was set aside for future use.

Categories. We selected a subset of the Reuters Region categories whose names exactly matched the names of geographical regions with entries in the CIA World Factbook (see below) and which had one or more positive examples in our large (23,149 document) training set. There were 189 such matches, from which we chose the 27 with names beginning with the letter A or B to work with, reserving the rest for future use.

[5]http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

Afghanistan
...
Geography
Location: Southern Asia, north of Pakistan
...
International disputes: periodic disputes with Iran over Helmand water rights; Iran supports clients in country, private Pakistani and Saudi sources also are active; power struggles among various groups for control of Kabul,
...
Government
Name of country:
conventional long form: Islamic State of Afghanistan
conventional short form: Afghanistan
...
Capital: Kabul

Figure 3: A portion of CIA WFB (1996 edition) entry for the category "Afghanistan".

Prior Knowledge. The domain knowledge text for each Region category was the corresponding entry in the CIA World Factbook 1996.[6] Figure 3 shows a portion of the entry for "Afghanistan". The HTML source code of the CIA WFB was downloaded in June 2004. The formatting of the entries did not make it easy to omit field names and boilerplate text. We instead simply deleted (in addition to HTML tags) all terms that occurred in 10% or more of the entries.

5.3 Text Representation

Text from each training and test document was converted to a sparse numeric vector in *SVM_Light* format (also used by BBR). The Bio Articles documents were in XML format. We concatenated the contents of the title (<*atl*>), subject (<*docsubj*>), and abstract (<*abs*>) elements and deleted all internal XML tags. For ModApte, we used the concatenation of text from the title (<*TITLE*>) and body (<*BODY*>) SGML elements of each article. For the RCV1 A-B Regions collection, we concatenated the contents of the headline (<*HEADLINE*>) and text (<*TEXT*>) XML element of each article.

For all datasets, text processing used the Lemur[7] utility ParseToFile. This performed case-folding, replaced punctuation with whitespace, and tokenized text at whitespace boundaries. The Lemur index files were then converted to document vectors in *SVM_Light* format.

In processing text for the Bio Articles and the ModApte datasets, the Porter stemmer [20] supplied by Lemur and the SMART [22] stoplist were used in conjunction with the Lemur utility ParseToFile.[8] For the RCV1-v2 dataset we used a convenient pre-existing set of document vectors we had prepared using Lemur without stemming or stopping. domain knowledge text corpora were processed in the same fashion as the corresponding task documents.

Within document weights were computed using cosine-normalized TFIDF weighting [23]. The initial weight of term

[6]http://www.umsl.edu/services/govdocs/wofact96/
[7]http://www-2.cs.cmu.edu/∼lemur
[8]ftp://ftp.cs.cornell.edu/pub/smart/english.stop or http://jmlr.csail.mit.edu/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

Dataset	Type of DK	DK Texts
Bio Articles	MeSH scope notes	22,995
ModApte Top 10	manually selected words	10
RCV1 A-B Regions	CIA WFB entries	189

Table 2: Type of domain knowledge texts and size of domain knowledge corpus for each of categorization datasets.

t_j in document d_i, w_{ij} was

$$w_{ij} = \begin{cases} (1 + log_e(f_{ij}))log_e \frac{N+1}{n_j+1}, & \text{if } t_j \text{ is present in } d_i, \\ 0, & \text{otherwise.} \end{cases}$$

Here N is the number of documents in the training population, f_{ij} is the frequency of term t_j in document d_i, and n_j is the number of training population documents containing term t_j. We use the *Lookahead IDF* variant of IDF weighting [6]. Cosine normalization was then applied to the TFIDF values.

5.4 Evaluation and Thresholding

We evaluated classification effectiveness using the F1 measure (harmonic mean of recall and precision) [14, 30], with macroaveraging (average of per-category F1 values) across categories. Both BBR and *SVM_Light* produce linear classifiers with thresholds intended to minimize error rate, so we retrained the thresholds to maximize observed F1 on the training data, while leaving other classifier parameters unchanged.

6. RESULTS

Our primary hypothesis was that using domain knowledge texts would greatly improve classifier effectiveness when few training examples are available, and not hurt effectiveness with large training sets. We also believed, given the diverse and non-document-like forms of the domain knowledge texts, that using them to specify prior distributions in a Bayesian framework was not only more natural, but more effective, than pretending they were additional training examples.

Table 2 summarizes the types of domain knowledge used, and the number of domain knowledge texts used to compute significance values for the Var/TFIDF and Mode/TFIDF methods. The number of categories used in the experiments was 32, 10 and 27 for the Bio Articles, ModApte and RCV1 collections, respectively.

6.1 Large Training Sets

This experiment trained classifiers on each collection's large training set. Table 3 presents macroaveraged F1 results for the three test collections. As found elsewhere [9], SVMs and lasso logistic regression show similar effectiveness, and both dominate ridge logistic regression. We note that our macroaveraged F1 for SVMs on ModApte Top 10 (86.55) is similar to that found by Wu & Srihari (approximately 83.5 on a non-random sample of 1,024 training examples, from the graph in Figure 3 [31]) and Joachims (82.5 with all 9,603 training examples, computed from his Figure 2 [11]).

Method DK Examples (using domain knowledge texts as artificial positive examples) had little impact on any learning algorithm with these large training sets. The four methods using prior probability distributions had little impact on lasso logistic regression, but gave a substantial benefit to ridge logistic regression on the two datasets with the lowest frequency categories.

6.2 Small Training Sets

The small training sets available in practical text classification situations are produced in a variety of unsystematic ways, making it hard to define what a "realistic" small training set is. We present results on three definitions that exhibit the range of properties we have seen using other definitions.

6.2.1 500 Random Examples

In this experiment we selected random training sets of 500 examples from the training population. The resulting training sets had 2 to 139 positive examples for categories in the Bio Articles collection, 9 to 184 positive examples for categories in the ModApte Top 10 collection, and 0 to 22 positive examples for categories in the RCV1 A-B Regions collection.

Table 4 provides the results. Effectiveness is lower than with large training sets, and the effect of the differing class frequencies is obvious. Lasso logistic regression is notably more effective on the small training sets than SVMs and ridge logistic regression. Method DK Examples gave improvements on two of three collections, but hurt the third. The Bayesian prior based methods, in contrast, always improved logistic regression results. For ridge logistic regression, the improvement was up to 1500%.

6.2.2 5 Positive and 5 Random Examples

Operational text classification tasks often originate with a handful of known positive examples. We simulated this by randomly selecting 5 positive examples of each class from the training population, and adding 5 additional examples randomly selected from the remainder without knowledge of class labels (Table 5). Since 5 positive examples is more than occurs in random samples of 500 examples for some classes, effectiveness is sometimes better and sometimes worse than in Table 4. Method DK Examples has a large impact with these tiny training sets, but the impact is sometimes good and sometimes bad. The prior based methods uniformly improve ridge regression (up to 130%) and usually improve lasso regression, though the risky Mode method hurts lasso substantially in two of the conditions.

6.2.3 5 Positive and 5 Closest Negative Examples

In a variation on the previous approach, we instead combined each of 5 random positive examples for each class with its nearest (based on highest dot product) negative neighbor. The theory was that someone attempting to quickly build a small training set might end up with positive and "near miss" examples. It is hard to know if this is true but, surprisingly, effectiveness (Table 6) was lower than when positives were supplemented with random examples (Table 5). In any case, we again see DK Examples having a large but unstable effect. The prior-based methods uniformly, sometimes greatly, improve ridge (up to 127%) and give small decrements (maximum 3.6%) to large improvements (maximum 79.7%) for lasso.

6.3 Analysis

	Bio Articles			ModApte Top 10			RCV1 A-B Regions		
Method	SVM	lasso	ridge	SVM	lasso	ridge	SVM	lasso	ridge
No DK	49.15	54.2	26.3	86.55	84.1	82.9	71.08	62.9	42.2
DK examples	50.55	54.4	26.8	86.55	84.3	82.1	71.09	64.2	42.3
Var		54.8	47.2		84.8	82.8		66.4	58.6
Var/TFIDF		55.2	52.2		84.6	83.8		70.8	68.9
Mode		53.2	35.3		84.2	82.7		59.2	47.1
Mode/TFIDF		53.3	41.9		83.6	83.1		64.5	62.9

Table 3: Macroaveraged F1 results for SVMs, lasso, and ridge logistic regression on three text categorization test collections using large training sets.

	Bio Articles			ModApte Top 10			RCV1 A-B Regions		
Method	SVM	lasso	ridge	SVM	lasso	ridge	SVM	lasso	ridge
No DK	9.06	35.1	2.6	69.24	72.5	37.6	8.45	23.1	3.3
DK examples	16.77	38.3	3.3	72.34	72.5	42.7	7.96	21.2	2.7
Var		44.5	34.4		74.8	73.1		32.9	23.0
Var/TFIDF		49.2	40.9		74.8	71.0		40.8	33.0
Mode		35.9	12.9		76.3	69.6		23.8	7.6
Mode/TFIDF		42.5	37.6		76.6	73.4		31.6	32.2

Table 4: Macroaveraged F1 results for SVMs, lasso, and ridge logistic regression on three text categorization test collections using 500 random examples in training sets.

	Bio Articles			ModApte Top 10			RCV1 A-B Regions		
Method	SVM	lasso	ridge	SVM	lasso	ridge	SVM	lasso	ridge
No DK	21.51	29.6	18.8	36.53	42.7	27.1	28.90	52.1	23.0
DK examples	17.78	41.0	11.9	34.52	61.2	22.3	39.29	47.2	38.7
Var		36.3	34.2		61.7	62.2		47.4	37.1
Var/TFIDF		34.3	35.7		61.3	61.5		50.7	53.0
Mode		23.7	24.0		57.1	62.2		34.7	27.2
Mode/TFIDF		36.4	33.9		58.5	62.1		51.5	48.8

Table 5: Macroaveraged F1 results for SVMs, lasso, and ridge logistic regression on three text categorization test collections using 5 positive and 5 random examples in training sets.

	Bio Articles			ModApte Top 10			RCV1 A-B Regions		
Method	SVM	lasso	ridge	SVM	lasso	ridge	SVM	lasso	ridge
No DK	19.87	21.4	18.8	33.41	34.4	33.0	21.84	30.6	23.0
DK examples	22.34	37.0	10.6	32.99	55.9	23.2	24.45	25.8	35.5
Var		30.5	31.9		34.0	60.4		37.4	37.3
Var/TFIDF		32.9	34.6		47.3	58.9		34.1	47.7
Mode		26.7	24.5		61.8	58.7		29.5	27.8
Mode/TFIDF		36.4	34.2		61.4	58.5		53.0	52.2

Table 6: Macroaveraged F1 results for SVMs, lasso, and ridge logistic regression on three text categorization test collections using 5 positive and their 5 closest negative examples in training sets.

Domain knowledge, in any form, generally had little effect with large training sets. The exception was ridge logistic regression, which was substantially improved on the two collections where some categories had few positives. Overall, ridge performed poorly given its popularity. A caveat is that many ModApte and RCV1 Regions categories have a dominant single predictor, a situation that favors lasso.

Treating domain texts as artificial training examples had an erratic impact, sometimes improving and sometimes substantially harming effectiveness. Converting domain texts to priors, on the other hand, almost always improved effectiveness (37 of 48 experimental conditions for lasso, and 48 of 48 for ridge from its poor baseline). As expected, mode-setting was risky, with method Mode proving either the best or, usually, the worst of the four prior setting methods 21 of 24 times. Where we had nontrivial domain corpus TFIDF weights (Bio Articles and RCV1 A-B Regions), they proved surprisingly useful. Var/TFIDF beat Var in 14 of 16 such conditions, and Mode/TFIDF beat Mode in 16 of 16. Other source of term quality information, such as stoplists or task-document IDFs, would likely prove useful as well.

Under a view that domain knowledge should do no harm we recommend either Var/TFIDF, which reduced effectiveness vs. No DK in only 1 of 24 conditions (by 2.7%), or Mode/TFIDF, which reduced effectiveness in only 3 of 24 conditions (by a maximum of 1.7%). Both usually provided large improvements.

7. SUMMARY AND FUTURE WORK

We have presented an initial, but highly effective, strategy for combining domain knowledge with supervised learning for text classification using Bayesian logistic regression. On three data sets, with three diverse sources of domain knowledge, we found large improvements in effectiveness, particularly when only small training sets are available. We

are continuing this work in many directions, including exploring the impact of variability in the choice of both small training sets and domain knowledge texts.

Beyond that, our research program is to recast many IR heuristics (stopword lists, stemming, term weighting, etc.) as appropriate priors, with the goal of using simple binary text representations and priors for which a somewhat sophisticated user could have meaningful numeric intuitions. Logistic regression is behind statements in medicine such as "eating food X increases you change of heart disease by Y%". It does not seem impossible to have similarly concrete prior knowledge of words in text classification.

Acknowledgements

The work was supported under funds provided by the KD-D group for a project at DIMACS on Monitoring Message Streams, funded through National Science Foundation grant EIA-0087022 to Rutgers University. The views expressed in this article are those of the authors, and do not necessarily represent the views of the sponsoring agency.

8. REFERENCES

[1] 2005. http://www.nlm.nih.gov/mesh.

[2] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *SIGIR '94*.

[3] B. Carlin and T. Louis. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall, London, 1996.

[4] K. Chai, H. Chieu, and H. Ng. Bayesian online classifiers for text classification and filtering. In *SIGIR '02*, pages 97–104, 2002.

[5] C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. In *EMNLP '04*, 2004.

[6] A. Dayanik, D. Fradkin, A. Genkin, P. Kantor, D. Lewis, D. Madigan, and V. Menkov. DIMACS at the TREC 2004 genomics track. In *TREC '04*, 2005.

[7] N. Fuhr and U. Pfeifer. Combining model-oriented and description-oriented approaches for probabilistic indexing. In *SIGIR '91*, pages 45–56, 1991.

[8] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *IJCAI '05*, pages 1048–1053, 2005.

[9] A. Genkin, D. Lewis, and D. Madigan. Large-scale bayesian logistic regression for text categorization. Technical report, DIMACS, 2004.

[10] W. Hersh, R. Bhuptiraju, L. Ross, A. Cohen, D. Kraemer, and P. Johnson. TREC 2004 genomics track overview. In *TREC '04*, 2004.

[11] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *ECML '98*, pages 137–142, 1998.

[12] T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer, 2002.

[13] R. Jones, A. McCallum, K. Nigam, and E. Riloff. Bootstrapping for text learning tasks. In *IJCAI '99 Workshop on Text Mining*, 1999.

[14] D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, April 2004.

[15] F. Li and Y. Yang. A loss function analysis for classification methods in text categorization. In *ICML '03*, pages 472–479, 2003.

[16] B. Liu, X. Li, W. Lee, and P. Yu. Text classification by labeling words. In *AAAI '04*, 2004.

[17] D. Madigan, J. Gavrin, and A. Raftery. Eliciting prior information to enhance the predictive performance of bayesian graphical models. *Communications in Statistics - Theory and Methods*, pages 2271–2292, 1995.

[18] T. Meyer and B. Whateley. SpamBayes: Effective open-source, Bayesian based, email classification system. In *CEAS '04*, 2004.

[19] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI'99 Workshop on Information Filtering*, 1999.

[20] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[21] H. Raghavan, O. Madani, and R. Jones. Interactive feature selection. In *IJCAI '05*, pages 841–846, 2005.

[22] G. Salton, editor. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, 1971.

[23] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *IPM*, 24(5):513–523, 1988.

[24] Gerard Salton and Michael J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[25] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *ICML '02*, 2002.

[26] H. Schutze, D. Hull, and J. Pedersen. A comparison of classifiers and document representations for the routing problem. In *SIGIR '95*, pages 229–237, 1995.

[27] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[28] R. Smith. Bayesian and frequentist approaches to parametric predictive inference (with discussion). In *Bayesian Statistics 6*. Oxford Univ. Press, 1999.

[29] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Statistical Soc. B.*, 58:267–288, 1996.

[30] C. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, London, 2nd edition edition, 1979.

[31] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *KDD '04*, pages 326 – 333, 2004.

[32] J. Zhang and Y. Yang. Robustness of regularized linear classification methods in text categorization. In *SIGIR'03*, pages 190–197, 2003.

[33] T. Zhang and F. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 4:5–31, 2001.