Learning from Labeled Features using Generalized Expectation Criteria

Gregory Druck
Dept. of Computer Science
Univ. of Massachusetts
Amherst, MA
gdruck@cs.umass.edu

Gideon Mann
Google, Inc.
76 9th Ave.
New York, NY
gideon.mann@gmail.com

Andrew McCallum
Dept. of Computer Science
Univ. of Massachusetts
Amherst, MA
mccallum@cs.umass.edu

ABSTRACT

It is difficult to apply machine learning to new domains because often we lack labeled problem instances. In this paper, we provide a solution to this problem that leverages domain knowledge in the form of affinities between input features and classes. For example, in a *baseball* vs. *hockey* text classification problem, even without any labeled data, we know that the presence of the word *puck* is a strong indicator of *hockey*. We refer to this type of domain knowledge as a *labeled feature*. In this paper, we propose a method for training discriminative probabilistic models with labeled features and unlabeled instances. Unlike previous approaches that use labeled features to create labeled pseudo-instances, we use labeled features directly to constrain the model's predictions on unlabeled instances. We express these soft constraints using generalized expectation (GE) criteria — terms in a parameter estimation objective function that express preferences on values of a model expectation. In this paper we train multinomial logistic regression models using GE criteria, but the method we develop is applicable to other discriminative probabilistic models. The complete objective function also includes a Gaussian prior on parameters, which encourages generalization by spreading parameter weight to unlabeled features. Experimental results on text classification data sets show that this method outperforms heuristic approaches to training classifiers with labeled features. Experiments with human annotators show that it is more beneficial to spend limited annotation time labeling features rather than labeling instances. For example, after only one minute of labeling features, we can achieve 80% accuracy on the *ibm* vs. *mac* text classification problem using GE-FL, whereas ten minutes labeling documents results in an accuracy of only 77%

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

SIGIR'08, July 20–24, 2008, Singapore.

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Learning with Domain Knowledge, Labeled Features, Semi-Supervised Learning, Text Classification

1. INTRODUCTION

Supervised machine learning methods require costly labeled problem instances, and this limits the applicability of learning to new domains. Semi-supervised learning methods [21], which aim to leverage available unlabeled instances, are an appealing solution for reducing labeling effort. However, despite recent interest in this problem, real applications of semi-supervised learning remain rare. Reasons for this may include the time and space complexity and reliance on sensitive hyperparameters of semi-supervised methods. Additionally, many methods make strong assumptions that may hold in small, synthetic data sets, but tend to be violated in real-world data.

Instead, we want a simple, robust method that facilitates training models for new domains and requires minimal annotation effort. One potential solution involves incorporating existing domain knowledge into learning. There has been much recent interest in this idea [20, 19, 9, 15, 17, 7]. In this paper, we propose a discriminative semi-supervised learning method that incorporates into training one particular type of domain knowledge: affinities between features and classes. For example, in a *baseball* vs. *hockey* text classification problem, even without any labeled data, we know that the presence of the word *puck* is a strong indicator of *hockey*. We refer to this type of domain knowledge as a *labeled feature*. Unlike previous heuristic approaches that use labeled features for feature selection or to create labeled pseudo-instances [6, 14, 17, 19, 20], we use labeled features directly to constrain the model's predictions on unlabeled instances. We specify these soft-constraints using generalized expectation (GE) criteria.

A GE criterion [16] is a term in a parameter estimation objective function that express preferences on values of a model expectation. GE is similar to the method of moments for parameter estimation, but allows us to express arbitrary scalar preferences on expectations of arbitrary functions, rather than requiring equality between sample and model moments. We also note three important differences from traditional training objective functions for undirected graphical models. (1) A one-to-one relationship between

GE terms and model factors is not required. For example, a GE term may score expectations on sets of variables that form a subset of parameterized model factors, or on sets of variables larger than model factors. (2) Model expectations in different GE terms can be conditioned on different data sets. (3) The score function can be arbitrary. Examples of possible score functions include the distance to some target expectation or a smooth hinge-loss.

In this paper, we use leverage property (3) to specify an objective function that penalizes parameter settings if the resulting model predictions do not conform to prior expectations. We use property (1) to express constraints only on the subsets of variables for which prior information is available. Specifically, for each labeled feature, there is a corresponding GE term that scores the model's predicted class distribution conditioned on the presence of that feature. The score function penalizes these distributions according to their KL-divergence from reference distributions estimated using the labeled features. We derive a specific objective function for a multinomial logistic regression classifier, but the idea is applicable to other discriminative probabilistic models. We refer to this method as Generalized Expectation with Feature Labels, or GE-FL.

We evaluate GE-FL on six text classification data sets. First, we show that GE-FL outperforms several baseline methods that use labeled features. Next, we compare with three previous methods that incorporate labeled features into learning [19, 20, 17], and show that GE-FL attains comparable or better performance despite using no labeled documents. Finally, we conduct human annotation experiments in which we compare the performance over time of (a) a system that trains a classifier with labeled features using GE-FL and (b) a system that uses semi-supervised training with labeled documents. The results show that given limited annotation time, it is more beneficial to spend that time labeling features rather than labeling instances. For example, after only one minute of labeling features, we can achieve 80% accuracy on the *ibm* vs. *mac* text classification problem using GE-FL, whereas ten minutes labeling documents results in an accuracy of only 77%. Given equal labeling time, the accuracy difference is often much more pronounced, with absolute accuracy improvements as high as 40%. In our experiments, labeling features is on average 3.7 times faster than labeling documents, a result that supports similar findings in previous work [18].

2. RELATED WORK

The methods described in this paper are semi-supervised [21]. However, the supervision comes in the form of labeled features, or more generally arbitrary expectations from domain knowledge, rather than labeled instances. We suggest that this approach is beneficial because it avoids some of the common assumptions of semi-supervised learning methods. For example, unlike discriminative semi-supervised learning methods such as Transductive Support Vector Machines [12] and Entropy Regularization [8], we do not assume low-density regions between classes.

There has been much recent interest in incorporating domain knowledge into learning, including several methods that use labels or relevance judgments on features. Nearly all of these methods convert labeled features into labeled instances, and apply a standard learning algorithm. Liu, et al. [14] use human annotators to label features that are highly predictive of unsupervised instance clustering assignments. The unlabeled instances are soft-labeled according to their cosine similarity with pseudo instances that only contain labeled features, and this soft-labeled data is used as an initialization point for the expectation maximization (EM) algorithm. Schapire, Rochery, and Gupta [19] use handcrafted rules based on relevant features to label instances, and modify AdaBoost to choose weak learners that both fit the labeled training data and the soft-labeled data. Wu and Srihari [20] use labeled features to assign labels and confidence scores to unlabeled instances, which are then used in conjunction with labeled data during training. We compare with the methods of Schapire, Rochery, and Gupta [19] and Wu and Srihari [20] in Sections 5.2 and 5.3, respectively. Dayanik, et al. [4] propose several methods that use labeled features to specify prior distributions on the parameters of a logistic regression model.

There is also recent work in the natural language processing community with similar goals. Chang, Ratinov, and Roth [2] propose an EM-like algorithm that incorporates prior constraints into semi-supervised training of structured output models. In the E-step, the inference procedure produces an N-best list of outputs ranked according to the sum of the output's score under the model and a penalty term for violated constraints. In the M-step, the N-best list is used to re-estimate the model parameters. Haghighi and Klein [9] use prototypes, which are analogous to what we refer to as labeled features, to learn log-linear models for structured output spaces. The prototypes are used to hypothesize additional soft prototypes for features that are syntactically similar. All prototypes are then used as features during maximum likelihood training on limited labeled data.

Other types of domain knowledge have also been incorporated into learning. Jin and Liu [11] and Mann and McCallum [15] provide methods for incorporating prior information about the class distribution into discriminative training. Huang and Mitchell [10] propose a new generative clustering model and provide methods for the user to exert influence over the learned clusters. For example, the user can specify that a feature indicates a cluster, an instance belongs to a cluster, or that a cluster should be deleted.

Many of the above methods convert domain knowledge into labeled instances. In this paper, we take an alternative approach in which domain knowledge is used to constrain model predictions. Graça, Ganchev, and Taskar [7] provide a related method that incorporates prior constraints into the EM algorithm. Specifically, the E-step is modified so that the expectation over output variables is the closest distribution (in terms of KL-divergence) to the model prediction that respects a specified set of constraints. In the M-step, the model parameters are re-estimated using this modified expectation. We note several differences between this method and GE-FL. First, the constraints in constrained EM are per-instance, whereas in this paper we use global constraints over entire data sets. Next, Graça et al. use a generative model, whereas here we use direct maximization in a discriminative model. Finally, Graça et al. put constraints only on the output variables, whereas here the constraints additionally consider input variables.

Work in active learning is also relevant. In active learning, the learner can choose the particular instances to be labeled. In pool-based active learning [3], the learner has access to a set of unlabeled instances, and can choose the instance that

has the highest expected utility according to some metric. A standard pool-based active learning method is uncertainty sampling [13], in which the instance chosen is the one for which the model predictions are most uncertain. Although in theory this method is problematic because it ignores the distribution over instances [5], in practice it often works well, and is easy to implement. We use uncertainty sampling as a baseline in our user experiments.

Some recent work has addressed active learning by labeling features. Raghavan, Madani, and Jones [18] interleave feedback on instances and features in an algorithm called tandem learning. They show that incorporating feedback on features can significantly accelerate active learning. Experiments also demonstrate that humans can provide accurate information about features, and that it takes five times as long to label instances as to label features. Raghavan and Allan [17] provide additional methods for training SVMs with labeled features, including scaling the parameters of labeled features, creating specially-weighted pseudo-instances containing only labeled features, and soft-labeling unlabeled instances. We compare with tandem learning in Section 5.4. Godbole et al. [6] describe software for interactive classification that uses both feature and instance active learning. Similarly to Raghavan and Allan [17], Godbole et al. incorporate information about features into training by creating pseudo-instances containing only labeled features.

3. GENERALIZED EXPECTATION CRITERIA

In this section, we describe Generalized Expectation criteria and derive the specific objective function we use to train classifiers with labeled features. Section 4 describes the process of obtaining labeled features and converting them into specific constraints.

A generalized expectation (GE) criterion is a term in a parameter estimation objective function that assigns scores to values of a model expectation [16]. In this paper we use GE in conjunction with discriminative probabilistic models. Given a score function S, an empirical distribution $\tilde{p}$, a function f, and a conditional model distribution p parameterized by θ, the value of a GE criterion is

$$S(E_{\tilde{p}(X)}[E_{p_\theta(Y|X)}[f(X,Y)]]).$$

One specific type of score function S is some measure of distance between the model expectation and a reference expectation. Given some distance function $\Delta(\cdot,\cdot)$, a reference expectation $\hat{f}$, an empirical distribution $\tilde{p}$, a function f, and a conditional model distribution p, this criterion is

$$\Delta(\hat{f}, E_{\tilde{p}(X)}[E_{p_\theta(Y|X)}[f(X,Y)]]).$$

In this paper, $\mathbf{x}$ is a vector of input feature counts, y is a discrete class label, and $p_\theta(y|\mathbf{x})$ is a conditionally trained Markov random field with a single output variable and observation variables that are conditionally independent given this output. The probability of output y conditioned on input $\mathbf{x}$ is given by

$$p_\theta(y|\mathbf{x}) = \frac{\exp(\sum_i \theta_{yi} x_i)}{Z(\mathbf{x})},$$

where $Z(\mathbf{x})$ is a normalizer that assures $\sum_y p_\theta(y|\mathbf{x}) = 1$. In the literature, this model is often referred to as multinomial logistic regression or a maximum entropy classifier.

We use GE terms in which $\tilde{p}$ is the distribution of unlabeled data U, and we compute the expectation of $f_k(\mathbf{x}, y) = \vec{I}(y)I(x_k > 0)$, an indicator of the presence of feature k in $\mathbf{x}$ times an indicator vector with 1 at the index corresponding to label y and zeros elsewhere. Therefore, $E_U[E_{p_\theta(y|\mathbf{x})}[f_k(\mathbf{x}, y)]]$ is a vector in which the ith value is the expected number of instances that contain feature k and have label y_i. If we additionally add a normalizing constant into f_k, $f_k(\mathbf{x}, y) = \frac{1}{C_k}\vec{I}(y)I(x_k > 0)$, where $C_k = \sum_{\mathbf{x}\in U} I(x_k > 0)$, the expectation is the predicted label distribution on the set of instances that contain feature k, $\tilde{p}_\theta(y|x_k{>}0)$. We use the KL divergence for $\Delta(\cdot,\cdot)$. A single term is then

$$\sum_y \hat{p}(y|x_k{>}0) \log \frac{\hat{p}(y|x_k{>}0)}{\tilde{p}_\theta(y|x_k{>}0)}, \quad (1)$$

where $\hat{p}(y|x_k > 0)$ are reference distributions obtained using domain knowledge. The estimation of reference distributions is discussed in Section 4. The combined objective function is composed of a GE term for each labeled feature $k \in K$, and a zero-mean σ^2-variance Gaussian prior on parameters.

$$\mathcal{O} = -\sum_{k\in K} D(\hat{p}(y|x_k{>}0))||\tilde{p}_\theta(y|x_k{>}0)) - \sum_j \frac{\theta_j^2}{2\sigma^2}$$

We use L-BFGS, a quasi-Newton optimization method, to estimate model parameters. The gradient of Equation 1 with respect to the model parameter for feature j and label y', $\theta_{y'j}$, is:

$$\begin{aligned}
&\frac{\partial}{\partial\theta_{y'j}} D(\hat{p}(y|x_k{>}0)||\tilde{p}_\theta(y|x_k{>}0)) \\
&= -\frac{\partial}{\partial\theta_{y'j}} \sum_y \hat{p}(y|x_k{>}0) \log \tilde{p}_\theta(y|x_k{>}0) \\
&= -\frac{1}{C_k} \sum_y \frac{\hat{p}(y|x_k{>}0)}{\tilde{p}_\theta(y|x_k{>}0)} \sum_{\mathbf{x}\in U} I(x_k{>}0) \frac{\partial}{\partial\theta_{y'j}} p_\theta(y|\mathbf{x}) \\
&= -\frac{1}{C_k} \sum_y \frac{\hat{p}(y|x_k{>}0)}{\tilde{p}_\theta(y|x_k{>}0)} \sum_{\mathbf{x}\in U} I(x_k{>}0) \\
&\quad \Big(I(y{=}y') p_\theta(y|\mathbf{x}) x_j - p_\theta(y|\mathbf{x}) p_\theta(y'|\mathbf{x}) x_j \Big) \\
&= -\frac{1}{C_k} \sum_y \frac{\hat{p}(y|x_k{>}0)}{\tilde{p}_\theta(y|x_k{>}0)} \sum_{\mathbf{x}\in U} p_\theta(y|\mathbf{x}) I(x_k{>}0) \\
&\quad \Big(I(y{=}y') x_j - p_\theta(y'|\mathbf{x}) x_j \Big)
\end{aligned}$$

Above, we observe that the degree to which the gradient of a parameter for an unlabeled feature j and label y' is affected by a GE-FL term for labeled feature k depends on how often j and k co-occur in an instance.

Because we only expect to have prior knowledge for a subset of features, there will be more parameters in the model than constraints in the objective functions. Consequently, we expect the optimization problem to be underconstrained, meaning that there will be many optimal parameter settings. Therefore, in practice we use GE in conjunction with other objective functions that help to choose among these possible models.

The Gaussian prior on parameters addresses this problem by preferring parameter settings with many small values over settings with a few large values. This encourages the

model to have non-zero values on parameters for unlabeled features that co-occur often with a labeled feature. That is, if the word *goal* occurs often in documents with *puck*, increasing the weight of *goal* can help to satisfy the constraint that the model should predict *hockey* conditioned on the presence of *puck*. The Gaussian prior prefers this setting, in which *puck* and *goal* both have moderate weights, to the setting in which *puck* has high weight and *goal* has zero weight, since it penalizes the square of the parameter values. We use this term in all experiments in this paper with $\sigma = 1$. Other terms that could help choose amongst possible models include standard conditional log-likelihood on labeled instances and agreement objective functions that encourage model predictions to be consistent when using different subsets of features.

4. LABELING FEATURES

In this section, we describe methods for selecting candidate features for labeling, obtaining labels for these features, and estimating the reference expectations needed for the KL divergence from target objective function.

4.1 Candidate Feature Selection

Oracle-features: Ideally, a selected feature should be both highly predictive of some class, and occur often enough to have a large impact. In practice we will not be able to determine whether a feature is predictive if we have no labeled instances. However, in order to obtain an upper bound on feature selection methods, we assume there exists an oracle that can reveal the label of each unlabeled instance. We then select features according to their predictive power as measured by the mutual information of the feature with the class label.

LDA-features: Another potential feature selection method would select features randomly only according to their frequency. The problem with this method is that it tends to select common, non-predictive features, such as stopwords in text classification. Instead we run unsupervised feature clustering and select the most prominent features in each cluster. In this paper we cluster unlabeled data with latent Dirichlet allocation (LDA) [1], a widely used topic model. For each LDA topic t_i, we sort features x_k by $p(x_k|t_i)$ and choose the top f features. There is no guarantee that the candidate features selected by this heuristic are relevant to the learning task of interest. However, in practice this performs much better than selecting candidate features by frequency.

For experiments in this paper, we choose the top $25L$ features according to these metrics, where L is the number of classes.

4.2 Obtaining Feature Labels

We first discuss the labeling process. When shown a candidate feature, the labeler can choose to accept the labeling request or discard the feature. The labeler only labels features that are accepted. Note that this process is different from traditional instance labeling because labeling requests may be refused. For example, if presented with the word "the", the labeler will likely discard it because it does not have strong affinity with any one particular label.

Oracle-labeler: For some experiments we use feature labels provided by an oracle rather than a human. To decide whether to accept a feature, the oracle is able to reveal the labels of the unlabeled instances in order to simulate human background knowledge of the relevance of the feature. Using the instance labels, the oracle computes the mutual information of the feature with the class label, and accepts if this mutual information is above a threshold α. In this paper, α is the mean of the mutual information scores of the top M most predictive features, where $M = 100L$, or 100 times the total number of labels. Note that a feature can be labeled with more than one class. If accepted, the oracle labels a feature with the class with which the feature occurs most often, and any other class that occurs with the feature at least half as often. We note that because M is typically small relative to the total number of input features, the oracle is somewhat conservative in the features it accepts. This simulates a scenario in which the user only knows about the most prominent and important features.

The second method for obtaining feature labels is to ask real annotators. We explore this approach in Section 6. For the experiments in Sections 5.2 and 5.3, we use labeled features provided in prior work.

4.3 Reference Distribution Estimation

Target or reference expectations are required by the KL divergence calculation. We present two methods for estimating reference expectations. We note that we could alternatively allow the users to specify the reference distributions directly during the labeling process. We choose not to do this because it is not clear that users can provide accurate estimates of these distributions. However, we could instead have the labeler specify a degree of association between a label and feature in terms of discrete categories such as *strongly indicative*. We plan to explore such approaches in future work, but note that the results in this paper seem to indicate that precise estimates of the reference distributions are not required to achieve good performance.

Schapire-distributions: As proposed by Schapire, et al. [19], we use a simple heuristic in which a majority of the probability mass for a feature is distributed uniformly among its associated classes(s), and the remaining probability mass is distributed uniformly among the other non-associated classes. Define q_{maj} as the probability for the associated classes. Then, if there are n associated classes out of L total classes, each associated class has probability $\hat{p}(y|x_k>0) = q_{maj}/n$ and each non-associated class has probability $\hat{p}(y|x_k>0) = (1-q_{maj})/(L-n)$. For the experiments in this paper, we use $q_{maj} = 0.9$.

Feature-voted-distributions: Alternatively, we use the labeled features to vote on labels for the unlabeled instances. For each feature x_k in an instance $\mathbf{x}$, it contributes a vote for each of its labels. We then normalize the vote totals to get a distribution over labels for each instance. With this soft-labeled data, we can estimate the reference distributions directly.

5. EXPERIMENTS

We evaluate the effectiveness of GE-FL on six text classification data sets. For all data sets, instances correspond to documents and features are word counts. For the tasks in which a single instance can be assigned multiple labels, we split the task into L one vs. all binary learning tasks, where L is the number of labels. For other data sets, we use multi-class classification. We describe the data sets below.

- **reuters21578:**[1] A standard text categorization data set in which task is to assign categories to news articles. We use the ModApte split and evaluate on the top 10 most frequent classes, as in [20] (9603 training instances, 3299 testing instances).
- **20 newsgroups**[2] The task is to classify messages according to the newsgroup to which they were posted. We use both the entire data set (20 classes, 20,000 instances) and binary subsets (2,000 instances).
- **movie**[3] The Polarity v2.0 data set, in which the task is to classify the sentiment of movie reviews as positive or negative (2,000 instances).
- **sraa**[2] The task is to classify messages about real and model automobiles and aviation with the appropriate newsgroup (4 classes, 73,218 instances).
- **webkb**[4] The task is to classify university webpages as *student*, *course*, *faculty*, or *project* (4,199 instances).
- **industry sector**[2] The task is to classify webpages according to a hierarchy of industrial sectors (4,582 instances). We use binary subsets, and the top level categories (7 classes).

For data sets without a standard test/train split, we randomly split the data such that 75% is used as training data, and the remaining 25% is reserved for testing. For the experiments in sections 5.1, 5.2, and 5.4 we use 10 such random splits and report the mean of the results. For experiments that do not use labeled instances we simulate unlabeled data by hiding labels of all instances. Experiments with GE-FL never include labeled instances.

5.1 Comparison with Baselines

We first compare GE-FL with several baseline methods, described below. For these experiments, we use the *oracle-labeler*.

- **feature voting:** Use the feature labels to vote on the classification.
- **feature labeling:** Use the feature labels to vote on labels for the unlabeled instances and train a supervised model on this data. We leave instances without labeled features unlabeled, and use hard class assignments, which provided significantly better results in our experiments.
- **labeled only:** Use GE to match reference distributions estimated from the labeled features, but disallow the use of unlabeled features.

We run experiments comparing the above baselines with GE-FL and provide the results in Tables 1 and 2. Datasets **med-space**, **ibm-mac**, and **baseball-hockey** are subsets of the **20 newsgroups** data set; **healthcare-financial** is a subset of the **industry sector** data set. The parenthesized number with each data set indicates the mean number of features labeled by the oracle labeler. The results presented in Table 1 are obtained using *oracle-features* and *Schapire-distributions*. This simulates a scenario in which there is a domain expert who can suggest and label relevant features. We also run experiments using *LDA-features* and *Schapire-distributions*, which simulates a scenario in which some candidate features are presented to the labeler. The results are presented in Table 2. GE-FL attains the highest macro-F1 in 7 of the 9 data sets using *oracle-features*, and 7 of 9 using *lda-features*. Results marked with a * indicate that GE-FL performs significantly better under a two-tailed paired t-test with $p = 0.05$.

We motivated GE-FL in terms of bootstrapping models for new domains, so we also perform experiments to determine the effectiveness of GE-FL in relation to semi-supervised training with labeled documents. To do this, we use entropy regularization [8], a discriminative semi-supervised learning method that aims to minimize the uncertainty of predictions on unlabeled data. This method introduces a tuning parameter λ that controls the weight of the regularizer relative to the data likelihood. We set $\lambda = 0.2$, a value that provided the best mean results across all data sets, and perform training with a deterministic annealing procedure. We report the number of instances at which the performance of GE-FL and the instance learning method are statistically indistinguishable. Raghavan, et al. [18] perform a thorough user study in which they conclude that it is five times faster to label a feature than to label a document. We use this result to present estimated speed-ups using GE-FL over entropy regularization. We note that in the computation of this estimated speed-up, we consider the number of features presented to the labeler, including those that are discarded. Since we expect discarding a feature to be faster than labeling a feature, the estimates in Table 2 are likely conservative.

Each of the baselines demonstrates an important point about GE-FL. **Feature voting** uses the domain knowledge only, whereas GE-FL uses this information to constrain model predictions on unlabeled data, and in the process learns about co-occurring features without labels. **Labeled only** demonstrates the importance of incorporating these co-occurring features without labels. Finally, **feature labeling** is equivalent to using the labeled features to infer constraints on all features, whereas GE-FL only specifies constraints on features that are known to be relevant.

5.2 Comparison with Schapire, Rochery, and Gupta [2002]

In this experiment, we compare GE-FL with boosting with prior knowledge [19]. Boosting with prior knowledge aims to maximize the conditional log likelihood of both labeled instances and instances classified using a hand-crafted model. The hand-crafted model classifies instances using the product of label probabilities for features, which are estimated from labeled features using the *Schapire-distributions* heuristic. Schapire et al. provide 138 labeled features for the **20 newsgroups** data set. For comparison, we use the same feature labels and use the *Schapire-distributions* heuristic to estimate reference distributions. We note that the experiments in [19] use n-gram features, whereas we use only unigram features. Comparing using the domain knowledge only, GE-FL gives approximately a 15% absolute error reduction from 64% error ([19] Figure 3) to 49% error. Furthermore, the boosting method requires the domain knowledge and between 400 and 800 labeled documents for boosting with prior knowledge to match the accuracy of GE-FL, which uses no labeled documents.

[1] `http://kdd.ics.uci.edu/`
[2] `http://www.cs.umass.edu/~mccallum/code-data.html`
[3] `http://www.cs.cornell.edu/People/pabo/movie-review-data/`
[4] `http://www.cs.cmu.edu/~webkb`

	Learing with Labeled Features				Labeled Instances Required	
data set	**feat. voting**	**feat. label**	**labeled only**	**GE-FL**	**sup. + ER**	**est. speed-up**
movie (43.7 of 50)	0.763*	0.766*	0.772*	**0.797**	150	15.0
sraa (97.5 of 100)	0.630*	0.596*	0.585*	**0.651**	160	8.0
webkb (88.8 of 100)	0.496*	0.477*	0.745*	**0.774**	70	3.5
med-space (50.0 of 50)	0.907*	0.932*	0.930*	**0.952**	90	9.0
ibm-mac (43.7 of 50)	0.853	**0.864**	0.861	0.855	110	11.0
baseball-hockey (50 of 50)	0.925*	0.927*	0.939*	**0.954**	200	20.0
20 newsgroups (494.4 of 500)	0.554*	0.560*	0.643*	**0.704**	650	6.5
financial-healthcare (50 of 50)	**0.653**	0.443*	0.539*	0.583	50	5.0
sector.top (163.9 of 175)	0.664*	0.657*	0.719*	**0.730**	140	4.0

Table 1: On the left, macro-averaged F1 for methods that use feature labels. Candidate features are selected using *oracle-features*. A * indicates that GE-FL performs significantly better using a two-tailed paired t-test, $p = 0.05$. On the right, the number of labeled instances at which semi-supervised training becomes statistically indistinguishable from GE-FL, and the estimated speed-up if labeling a feature is 5 times faster than labeling a document.

	Learing with Labeled Features				Labeled Instances Required	
data set	**feat. voting**	**feat. label**	**labeled only**	**GE-FL**	**sup. + ER**	**est. speed-up**
movie (4.6 of 50)	0.616	0.608	0.607*	**0.623**	20	2.0
sraa (29.5 of 100)	**0.577**	0.526*	0.520*	0.559	80	4.0
webkb (17.5 of 100)	0.514*	0.513*	0.593*	**0.615**	20	1.0
med-space (14.3 of 50)	0.857*	0.862*	0.867*	**0.927**	40	4.0
ibm-mac (10.4 of 50)	0.740*	**0.817**	0.762*	**0.817**	50	5.0
baseball-hockey (10.8 of 50)	0.779*	0.840*	0.853*	**0.915**	40	4.0
20 newsgroups (269.6 of 500)	0.493*	0.514*	0.585*	**0.667**	300	3.0
financial-healthcare (9.4 of 50)	0.552*	0.456*	**0.595**	0.588	50	5.0
sector.top (50.7 of 175)	0.538*	0.534*	0.544*	**0.596**	60	1.7

Table 2: Same as above, but candidate features are selected using *lda-features*.

5.3 Comparison with Wu and Srihari [2004]

Next, we compare GE-FL with a method for leveraging labeled features using Weighted Margin Support Vector Machines (WMSVMs) [20]. Wu and Srihari provide a few features associated with each of the top 10 most frequent classes in the ModApte split of the **Reuters21578** data set. With WMSVMs, a macro-average break-even-point of around 0.53 is obtained using only this domain knowledge, and a macro-average break-even-point of around 0.60 is obtained using domain knowledge and 16 labeled examples ([20] Figure 3). Using the same domain knowledge, *feature-voted-distributions*, and no labeled documents, GE-FL attains a break-even-point of 0.630.

5.4 Comparison with Raghavan [2007]

We also provide an informal comparison with tandem learning [17], an active learning algorithm that incorporates feedback on instances and features into learning with Support Vector Machines. We call the comparison informal because tandem learning is quite different from GE-FL. Importantly, GE-FL uses neither active learning nor labeled documents. In the referenced experiments, tandem learning uses a total of 12 labeled documents, and shows at most 100 features to the annotator. Both features and instances are actively selected to reduce uncertainty. Conversely, we use a static list of features, chosen before learning begins using unsupervised clustering. We compare performance on the **20 newsgroups** data set. We use a one vs. all setup for better comparison. Raghavan et al. report macro-F1 of 0.354 ([17] Table 3). With 100 candidate features selected using *lda-features*, reference distributions estimated using *association-voted-distributions*, and the *oracle-labeler*, we attain macro-F1 of 0.477, averaged over 10 random splits of the data. This result is encouraging because it suggests that combining GE-FL with active feature learning could produce even better results.

6. USER EXPERIMENTS

Finally, we conduct annotation experiments in which we time three users as they label 100 documents and 100 features for binary classification tasks. The candidate features are selected using *lda-features*. The features are presented one at a time, and the user can choose a single label for the feature or choose to discard the feature. After the users finish labeling features, they label documents, again with the option to choose a label for the document or to ignore the document if it appears ambiguous. We prefer this ordering (labeling features followed by documents) in order to give maximum benefit to the traditional document labeling method. We choose documents to present to the user with uncertainty sampling: after each instance is labeled, the instance with the most uncertain classification under the current model is selected next for labeling. We do this to ensure that the instances chosen for labeling are beneficial. The list of candidate features is static.

First, we are interested in the accuracy of the human annotators. Table 3 shows the labeling precision and recall for different annotators. For feature labeling, performance is measured using the oracle labeler as ground truth; for document labeling, performance is measured using the true

	doc. labeling		feat. labeling	
user + dataset	prec	rec	prec	rec
1 ibm-mac	0.90	0.58	0.80	1.00
1 med-space	0.95	0.86	0.73	1.00
1 baseball-hockey	0.98	0.84	0.52	0.92
2 ibm-mac	0.92	0.37	0.50	0.80
2 med-space	0.98	0.80	0.52	0.96
2 baseball-hockey	0.96	0.71	0.41	1.00
3 ibm-mac	0.91	0.75	0.86	1.00
3 med-space	0.99	0.75	0.67	1.00
3 baseball-hockey	0.96	0.83	0.54	1.00
Overall mean	0.95	0.72	0.62	0.96

Table 3: User labeling performance with respect to the oracle.

med: blood, cancer, care, disease, doctor, doctors, drugs, health, medical, medicine, pain, patients, vitamin, yeast **space**: earth, launch, mars, mission, moon, nasa, orbit, planet, satellite, shuttle, sky, space, universe
ibm: hp, dos, ibm **mac**: apple, mac
baseball: ball, baseball, braves, cubs, hit, hitter, jays, pitching, runs **hockey**: flyers, goal, hockey, leafs, nhl, period, shots

Table 4: Features that all three users labeled.

labels. The labelers provided precise labels for documents, but also discarded many documents. Conversely, the labelers were able to correctly label most features that the oracle considers relevant, but often also labeled other features. Inspection of these other features indicates that they are in fact moderately relevant. We defined the oracle to be conservative when labeling features, only choosing features that are almost certainly relevant. These results indicate that we may be able to allow the oracle to be less discerning in future work and perhaps further increase accuracy. User 2 had the most trouble selecting and labeling features. We suspect that this indicates insufficient familiarity with the learning tasks. This suggests that future experiments should involve an opportunity to look through the data before annotation. However, it does not seem unreasonable to assume that the annotators are familiar with the task they are trying to solve.

Figure 1 shows the accuracy of two trained systems over time. The first uses the labeled features and unlabeled instances with GE-FL. Reference distributions are estimated using *Schapire-distributions* with $q_{maj} = 0.9$. The second uses entropy regularization (ER) [8] (in this experiment we use direct maximization and weighting parameter $\gamma = 0.01$) with the labeled and unlabeled instances. Annotating features yields large accuracy improvements for the same amount of time. On average across all experiments, labeling features is 3.7 times faster than labeling documents, and the models trained with GE-FL have 1.0% higher final accuracy. Note that the point at which the GE-FL curve changes from a dotted line into dots indicates the point at which the user had processed all 100 features.

When the annotator is accurate, the results with feature labeling can be quite striking. For example, consider the results of User 1 for the **ibm vs. mac** classification task. The accuracy of the GE-FL system after 30 seconds of feature labeling is better than the accuracy of the ER system after 12 minutes of document labeling, a 24x speed-up. As another example, User 3 achieves accuracy of 90% on the **baseball vs. hockey** task after 90 seconds with the GE-FL system, at which point the ER system accuracy is around 50%.

Notice that the ER system gives erratic performance, with large accuracy jumps in consecutive 30 second intervals. This reinforces our earlier assertions about the brittleness of current semi-supervised methods.

7. CONCLUSION AND FUTURE WORK

In this paper, we have contributed GE-FL, a method for learning discriminative probabilistic models from labeled features and unlabeled documents. In experiments on text classification data sets this method outperforms heuristic methods that leverage labeled features. A preliminary user study supports the claim made in previous work [18] that it is much faster to label a feature than an instance. Consequently, GE-FL can provide dramatic decreases in the amount of time needed to train a classifier for a new domain.

In ongoing research, we are applying GE to models for structured output spaces and to the problems of active learning and domain adaptation. We are also interested in incorporating domain knowledge from ontologies and existing resources, and encoding task-specific structural constraints on the learning problem.

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by DoD contract #HM1582-06-1-2013, and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

9. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] M. Chang, L. Ratinov, and D. Roth. Guiding semi-supervision with constraint-driven learning. In *ACL*, 2007.

[3] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[4] A. Dayanik, D. D. Lewis, D. Madigan, V. Menkov, and A. Genkin. Constructing informative prior distributions from domain knowledge in text classification. In *SIGIR*, pages 493–500, 2006.

[5] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[6] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. Document classification through interactive supervision of document and term labels. In *PKDD*, pages 185–196, 2004.

[7] J. Graca, K. Ganchev, and B. Taskar. Expectation maximization and posterior constraints. In *NIPS*, 2007.

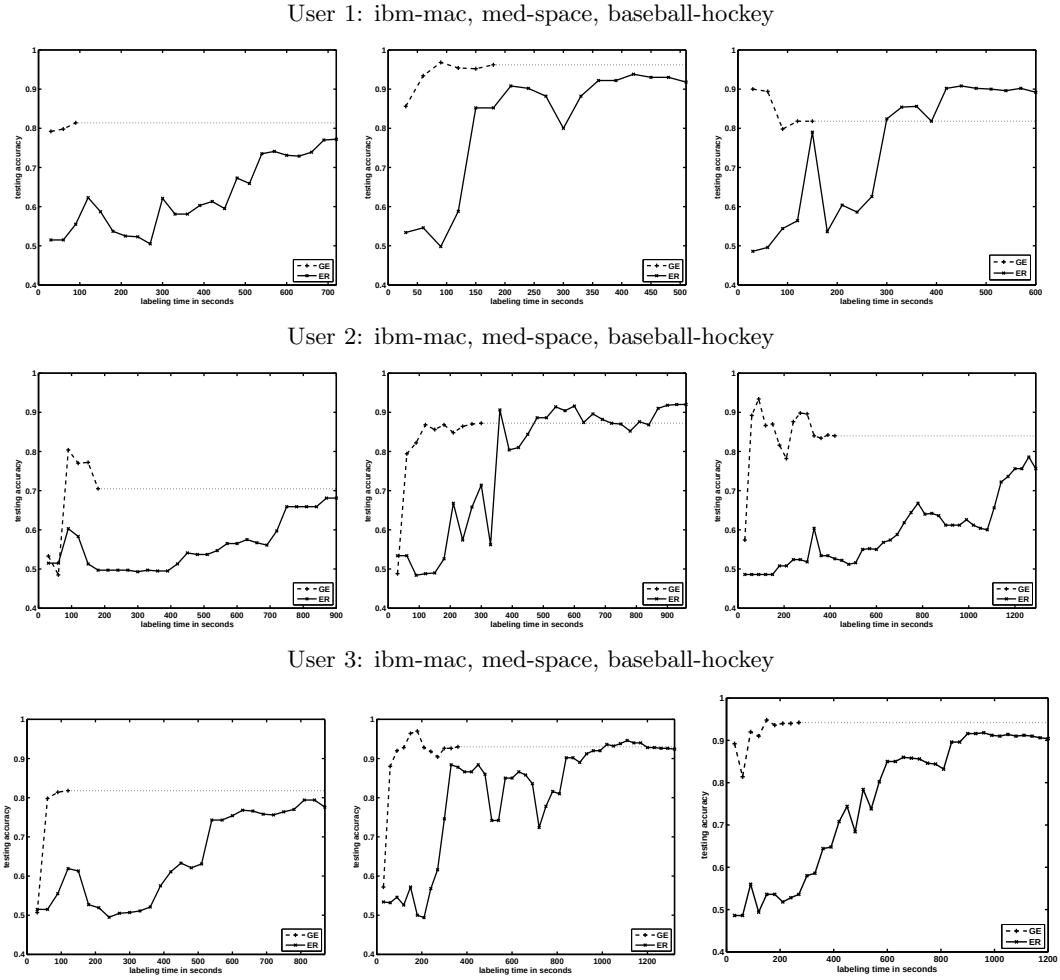

Figure 1: Accuracy vs. time for the GE-FL and ER systems. In most cases, GE-FL gives better accuracy given the same amount of annotation time.

[8] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004.

[9] A. Haghighi and D. Klein. Prototype-driver learning for sequence models. In *NAACL*, 2006.

[10] Y. Huang and T. M. Mitchell. Text clustering with extended user feedback. In *SIGIR*, pages 413–420, 2006.

[11] R. Jin and Y. Liu. A framework for incorporating class priors into discriminative classification. In *PAKDD*, 2005.

[12] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[13] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *ICML*, 1994.

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

[15] G. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML*, 2007.

[16] A. McCallum, G. Mann, and G. Druck. Generalized expectation criteria. Technical Report 2007-62, University of Massachusetts, Amherst, 2007.

[17] H. Raghavan and J. Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *SIGIR*, pages 79–86, 2007.

[18] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.

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

[20] X. Wu and R. K. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *SIGKDD*, 2004.

[21] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.