Journal of Machine Learning Research () Submitted 7/08; Published

Generalized Expectation Criteria for Semi-Supervised Learning with Weakly Labeled Data

Gideon S. Mann* GIDEON.MANN@GMAIL.COM
Google Inc.
76 9th Avenue
New York, NY 10011

Andrew McCallum MCCALLUM@CS.UMASS.EDU
Department of Computer Science
140 Governors Drive
University of Massachusetts, Amherst, 01003

Editor:

Abstract

In this paper, we present an overview of *generalized expectation criteria (GE)*, a simple, robust, scalable method for semi-supervised training using weakly-labeled data. GE fits model parameters by favoring models that match certain expectation constraints, such as marginal label distributions, on the unlabeled data. This paper shows how to apply generalized expectation criteria to two classes of parametric models: maximum entropy models and conditional random fields. Experimental results demonstrate accuracy improvements over supervised training and a number of other state-of-the-art semi-supervised learning methods for these models.

Keywords: Generalized Expectation Criteria, Semi-Supervised Learning, Logistic Regression, Conditional Random Fields

1. Introduction

Semi-supervised learning, where a small amount of human annotation is combined with a large amount of unlabeled data to yield an accurate classifier, has received a significant amount of attention from the research community. However, there are surprisingly few cases of its use in applications, where the emphasis is on solving a task, not on advancing theoretical understanding. This may be partially due to the natural time it takes for new machine learning ideas to propagate to practitioners, but we believe it is also due in large part to the inherent difficulty of the task and the unreliability of existing methods.

Instead of addressing the difficulties of semi-supervised learning head-on, we propose to use weakly labeled data ("side-information") in semi-supervised learning. To use this data, we present *generalized expectation critera (GE)*, a method initially described as *expectation regularization* in Mann and McCallum (2007). GE represents a new family of semi-supervised learning, where models are fit by minimizing model divergence from an input distribution. To calculate the divergence from the input distribution there is no need for additional training data, as the expected distribution on the unlabeled data can be used. These terms can be easily integrated with other terms, such as traditional log-likehood.

The experiments in this paper explore an illustrative special case of GE, *label regularization*, where a marginal distribution over output labels is applied as an expectation constraint. We investigate two parametric models: maximum entropy models and their structured output analog, conditional random fields. We demonstrate that for both of these models, label regularization is able to provide performance gains over other supervised and semi-supervised learning methods.

Generalized expectation criteria have a number of advantages over alternative semi-supervised learning techniques that make it suitable for use in practice. It is simple, making it easy to implement and use. It requires no additional processing such as constructing an inverted index for graph construction or pre-clustering unlabeled data. Since it can handle a wide spectrum of side-information, human intuitions about the problem can be explicitly communicated to the learning process, in contrast to other methods which can require opaque supervision, such as carefully tuned initialization, or specification of "contrastive examples" and other "auxiliary functions". We apply GE in this paper to discriminative models, and thus it is able to robustly handle overlapping, non-independent feature sets and yields transparent confidence estimates (in the form of probabilities). Additionally, GE has all of the advantages of parametric models, in particular scalability and a small memory footprint at test time.

2. Related Work

Traditional *supervised learning* takes as input fully labeled data, a set of tuples $\mathcal{D} = \{(x, y)\}$, where x is the input and y the desired output. The learner is a function which maps the input to a predictive function: $J(\mathcal{D}) = f$, where $f : x \rightarrow y$. Supervised learning is powerful, but the amount of labeled data needed can require significant human time and effort to create. In an effort to reduce the need for human effort, the machine learning community has explored semi-supervised learning. In *semi-supervised learning* approaches, a small amount of labeled data is augmented by unlabeled data, a set of elements $\mathcal{U} = \{x\}$, which it exploits to chose a similar function: $J(\mathcal{D} \cup \mathcal{U}) = f$. This section presents the main methods for semi-supervised learning from labeled and unlabeled data: 1) bootstrapping, 2) expectation maximization, 3) feature discovery, 4) decision boundaries in sparse region methods, and 5) graph-based methods.

In contrast to these methods, GE criteria exploit semi-supervised learning from weakly labeled data. With this scenario, there be no labeled data. Instead, in addition to unlabeled data there is side information that has been provided to the learner, for example, expectation constraints like marginal label distributions $\tilde{g}_y = p(y)$[1]. Section 2.2 reviews prior work in this area.

2.1 Semi-supervised Learning with Labeled and Unlabeled Instances

There are five main prior categories of semi-supervised learning approaches: bootstrapping, expectation maximization, feature discovery, decision boundaries in sparse regions, and graph-based methods.

Bootstrapping

In bootstrapping, or self-training approaches, a classifier is first trained on the fully labeled instances and then is applied to unlabeled instances. Some subset of those newly labeled instances are then used (in conjunction with the original labeled instances) to retrain the model.

1. Liang et al. (2009) presents a taxonomy of side-information.

Algorithm 1 Bootstrapping for semi-supervised learning

$f^{(0)} \leftarrow J(\mathcal{D})$
repeat
  $\mathcal{U}_B \leftarrow \cup_{x_i \in \mathcal{U}}(x_i, f^{(t-1)}(x_i))$
  $f^{(t)} \leftarrow J(\mathcal{D} \cup \mathcal{U}_B)$
until done

One of the most successful examples of this work is Yarowsky (1995), where a small 'seed set' of labeled instances is incrementally augmented. Co-training Blum and Mitchell (1998) looks at the case where two complementary classifiers can both be applied to a particular problem. Abney (2004) provides a deeper understanding of these methods by demonstrating that they optimize a natural objective function. However, these methods typically require continual human intervention in order to avoid performance loss during the bootstrapping process, such as in Riloff and Shepherd (2000).

Expectation Maximization

Generative models trained by expectation maximization (Dempster et al., 1977) have been widely studied for semi-supervised learning. EM consists of two steps: an expectation step $Q(\theta|\theta^{(t)}) = E_{p(y|x,\theta^{(t)}}[\log L(\theta; x, y)]$, and a maximization step, $\theta^{(t+1)} = \texttt{argmax}_\theta\ Q(\theta|\theta^{(t)})$. To apply EM to semi-supervised data, the log-likelihood function, $\log\ L(\theta; x, y)$, is set to be a composite of labeled and unlabeled data (possibly with a weighting factor to down-weight the contribution to the likelihood from the unlabeled data). One popular example of the use of EM for generative models is Nigam et al. (1998) which presents a naïve Bayes model for text classification trained using EM and semi-supervised data.

EM has also been applied to structured classification problems such as part-of-speech tagging (Klein and Manning, 2004), where EM can succeed after very careful and clever initialization. While these models can often be very effective, especially when used with "prototypes" (Haghighi and Klein, 2006b), they cannot efficiently accommodate non-independent features, e.g. those that span multiple inputs. In these cases, the dynamic program required to compute the feature expectations over all input positions quickly becomes intractable, as building the lattice requires exponential space in the length of input features.

EM for discriminative models has also been explored. Wang et al. (2002) proposes a EM based model which instead of the likelihood, maximizes the entropy given the latent variables. Alternatively, Salakhutdinov et al. (2003) present an expected gradient method, which can be used for discriminative models which have some partially observed labels, and McCallum et al. (2005) uses this method for conditionally trained CRFs. While this method is appealing in the case of input data that consists of sequences with partially hidden variables, it cannot be applied to scenarios with fully unlabeled instances.

Another twist on this technique is to blend generative and discriminative models, by combining ML estimates over the labeled data with EM parameter estimates over the unlabeled data, for a joint model which combines a CRF and a HMM (Suzuki et al., 2007; Suzuki and Isozaki, 2008). In this formulation, the log-likelihood can be viewed as two separate log-likelihood functions $L_1(\theta)$ and $L_2(\theta)$ which respectively correspond to the CRF and HMM log-likelihood. When optimizing

$L_1(\theta)$ (using maximum likelihood), the HMM parameters are held constant, and the reverse when optimizing $L_2(\theta)$.

While EM can sometimes works well, it is often fragile and finds solutions that are worse than the equivalent supervised model. Merialdo (1994) gives a classic example where EM fails to help part-of-speech tagging. Cozman and Cohen (2006) discuss the risks of using EM and describe situations where it can fail. Additionally, the generative models on which EM depends often perform worse than discriminative models.

Feature Discovery

As alternative to estimating a classifier directly with unlabeled data, a number of groups have explored using the unlabeled data for feature induction or feature discovery, and those features are then embedded into a traditional supervised learning problem. A latent clustering is applied over the unlabeled data, to learn a function f_l, which is used to provide additional features $f_l(x) = z$, these features are then used to augment the original labeled data: $\bar{\mathcal{D}} = \cup_{(x_d,y_d)\in\mathcal{D}}(x_d \cup z_d, y_d)$, and then supervised learning proceeds as usual. For example, (Miller et al., 2004; Ganchev et al., 2007) apply the method described in Brown et al. (1992) to cluster all of the word tokens in a large unsupervised corpus. Then for a given sentence, in addition to standard features, additional features corresponding to the latent clusters of the tokens in the sentence, are added. This technique, along with similar approachs (Freitag, 2004; Li and McCallum, 2005), have yielded small but consistent success. This method can be applied independently of the particular training method and in Section 6.3, we explore combining our method with those described by Miller et al. (2004).

Ando and Zhang (2005) use a similar method, but the clustering they explore is composed of auxiliary problems (e.g. predict a given token given the token context). In their method, they estimate the parameters for a linear classifier for each auxiliary problem, and then these parameters are embedded as a transformation of the parameters for a linear classifier for the original problem. Although this method has produced impressive gains, it is quite sensitive to the selection of auxiliary information, and making good selections requires significant insight. F. Pereira and J. Blitzer.[2] note that the list of tricks necessary to get the method of (Ando and Zhang, 2005) to work includes: oversampling positive instances, selecing the unlabeled data carefully, scaling real-valued features, and choosing the appropriate feature types.

Additionally, feature discovery as a semi-supervised learning technique relies on having a substantial amount of labeled data for training. It cannot be used in cases where only a limited amount of labeled data is available.

Decision Boundaries in Sparse Regions

Another family of methods uses the intuition that decision boundaries ought to fall in low-density regions (corresponding to an assumption of class separability) and thus fit discriminative models with this objective in mind. Clearly, if the cluster assumption is violated (*i.e.* the classes are not widely separable), assigning decision boundaries to low density regions is a poor choice. One illustrative example is entropy regularization (Grandvalet and Bengio, 2004), where a traditional conditional label log-likelihood objective function is augmented with an additional term that minimizes the

2. Personal communication.

entropy of the label distribution on the unlabeled data:

$$\begin{aligned} O(\theta; \mathcal{D}, \mathcal{U}) &= L(\theta; \mathcal{D}) + H(\theta; \mathcal{U}) \\ &= \sum_{d \in \mathcal{D}} \log p(y_d | x_d) - \lambda \sum_{u \in \mathcal{U}} \sum_{y} p(y | x_u) \log p(y | x_u) \,. \end{aligned}$$

This objective function favors parameter settings where the model is certain of the labels on given unlabeled data. In entropy regularization, the hyper-parameter λ has a dramatic effect on the performance of the learner, since it must be tuned with regards to the amount of labeled and unlabeled data. Entropy regularization is particularly difficult to apply in cases of very small amounts of labeled data, since in one degenerate case, the model could select one output label for all possible inputs. Studies on structured output models in Jiao et al. (2006) experimentally demonstrate that careful tuning of λ is mandatory.

Transductive support vector machines (TSVMs) (Joachims, 1999) add a constraint to the SVM optimization function in order to preserve the margin over unknown test labels:

$$(\{y_u\}, \theta) = \underset{\{y_u\}, \theta}{\texttt{argmin}} \; \frac{1}{2} ||\theta||^2 \quad \textit{subject to} \begin{cases} \forall_{x_i \in \mathcal{D}} & y_i [\theta \cdot x_i + b] \geq 1 \\ \forall_{x_u \in \mathcal{D}} & y_u [\theta \cdot x_u + b] \geq 1 \end{cases} .$$

It is combinatorially intractable to do a brute-force search over all possible labelings $\{y_u\}$, so an approximation search must be undertaken. Even with these approximations, the algorithm as originally proposed has running time $O(n^3)$. Sindhwani and Keerthi (2006) propose a method for speeding up training in some cases. In our experience, like entropy regularization, TSVMs also require extensive and delicate tuning of meta-parameters. We note that Sindhwani and Keerthi (2006) report results with meta-parameters tuned on test data. Benchmark tests have shown that entropy regularization performs as well as TSVMs (when the SVM is given a linear kernel) (Chapelle et al., 2006). Another related method is information regularization (Corduneanu and Jaakkola, 2003), which measures distance via the mutual information between a classifier and the marginal distribution $p(x)$.

Graph-based Methods

Graph-based (manifold) methods can be very accurate when applied to semi-supervised learning. In these methods, a graph, typically with weighted edges, is constructed spanning the labeled and unlabeled instances. Thereafter, unlabeled instances are assigned labels according to their neighbors. Zhu and Ghahramani (2002) propose label propagation, where labels propagate from labeled instances to unlabeled instances (see Algorithm 2). In this formulation, there are two significant

Algorithm 2 Label propagation

repeat

$\forall_{x_u \in \mathcal{U}} : \; p(y_u^{(t)} | x_u^{(t)}) = \frac{1}{Z} \sum_{j \in \mathcal{N}(x_u)} q(j \rightarrow u) p(y_j^{(t-1)} | x_j)$

until $p(y_u^{(t)} | x_u^{(t)})$ converges

choices that must be made: the graph structure (the neighborhood $\mathcal{N}(x)$ for each instance) and the transition function $q(j \rightarrow i)$. Szummer and Jaakkola (2002) present a closely related approach which uses random walks through the graph to assign labels. More distantly related, Li

and McCallum (2004) examine a method which performs an implicit clustering over points, as it simultaneously estimates pair-wise distance and classification boundaries.

As originally proposed, graph-based methods are slow, requiring time $O(n^3)$ or on average $O(kn^2)$ where k is the number of neighbors (similar to TSVMs). By sub-sampling unlabeled data, one can reduce run-time from $O(n^3)$ to $O(m^2n)$, where m is the subsampled number of unlabeled data points (Delalleau et al., 2006), but subsampling does not take full advantage of available unlabeled data. Zhu and Lafferty (2005) propose alternative methods for reducing the time compleity to $O(m^3), m < n$, but these may also impact performance. For structured output spaces, Lafferty et al. (2004) and Altun et al. (2005) have looked at approaches using these methods. However, the high running time of these methods has prevented wide-scale adoption, and they have been tested predominantly on synthetic or toy examples (e.g. with 5 labeled examples). Recently, Baluja et al. (2008) proposed a method for performing graph-based semi-supervised learning in parallel.

Since these are non-parametric models, they do not build a compact encoding of the model, and so it is not always clear how to apply them inductively (on new unlabeled data). At the very least, labeled and unlabeled data must be stored in order to classify new examples. In this paper we compare against a representative graph-based label propagation method called Quadratic Cost Criterion (QC) (Bengio et al., 2006) whose results are reported in Chapelle et al. (2006).

Difficult Applications

There are cases of semi-supervised learning being used in application settings, however, not without difficulty. In fact, a broad survey of semi-supervised learning methods (Chapelle et al., 2006) found that they do not uniformly beat supervised methods and that there is no clear winner from among the methods. This conclusion reflects the experimental evidence and theoretical support from a large span of work.

Expectation-maximization is notoriously fickle for semi-supervised learning. In a classic result Merialdo (1994) attempts semi-supervised learning to improve HMM part-of-speech tagging and finds that EM with unlabeled data reduces accuracy. Ng and Cardie (2003) also apply EM but finds that it fails to improve performance, as do Grenager et al. (2005) (without their tricky initialization). Cozman and Cohen (2006) discuss use cases where EM might fail to work.

Krogel and Scheffer (2004) use transductive SVMs for the functional genomics KDD Cup challenge and find that not only does it fails to improve performance but it even deteriorates performance. Ifrim and Weikum (2006) also find that TSVMs deteriorate performance. Kockelkorn et al. (2003) use transductive SVMs for text classification, but complain that it is computationally costly. Zhang and Oles (2000) discuss theoretical reasons why TSVMs might fail to work in various scenarios.

Macskassy and Provost (2006) apply harmonic mixing to classification of relational data, however the running time of harmonic mixing proves to be a barrier to its use. In the case of word sense disambiguation, Niu et al. (2005) has looked at label propagation, and found that the metric for graph construction has a dramatic effect on performance. Chen et al. (2005) look at combining manifold methods (e.g. ISOMAP) with semi-supervised learning, but finds that the methods are too fragile in their tuning parameters to be effective. Blum and Chawla (2001) also cite fragility in the tuning parameters as a problem for their graph-based method.

2.2 Semi-supervised Learning with Weakly Labeled Data

As an alternative to semi-supervised learning with labeled and unlabeled data, a number of methods have investigated semi-supervised learning with weakly labeled data or side information, though none with the expressiveness in labeling allowed by GE criteria.

Graph-based methods have used class proportions for post-processing to set thresholds on label propagation (Zhu et al., 2003). Schuurmans (1997) uses predicted label distributions on unlabeled data for model structure selection (as opposed to parameter estimation). More distantly, conditional harmonic mixing (Burges and Platt, 2006) minimizes over each point the KL-divergence between the currently predicted label distribution and the distribution predicted by its neighbors. Wang et al. (2004) also look at methods for incorporating class proportions into classification. In their model, they pseudolabel instances and provide them as constraints for the model to handle.

The use of side information to train a parametric classifier has been explored before by Schapire et al. (2002) who uses a boosted ensemble of weak learners set from human-generated expected distributions. There are significant differences between GE and this work, in particular, Schapire et al. (2002) match distributions on a per-instance basis, while generalized expectation criteria match a global distribution. Thus in the model proposed by Schapire et al. (2002), every example has to match the distribution given as input[3]. Graca et al. (2008) integrates similar types of instance-based constraints into EM learning, where the constraints restrict the space over which the model calculates the expectations of the hidden variables.

Like Schapire et al. (2002), Jin and Liu (2005) present a model for incorporating class proportions into discriminative models which places a expected class distribution over each instance. Unlike Schapire et al. (2002), these distributions start from a fixed point and then are allowed to change during training.

In contrastive estimation (Smith and Eisner, 2005), EM is performed over a restricted log-likelihood function, where instead of $L(\theta) = \sum_i \log p(x_i; \theta)$, the contrastive estimation log-likelihood function is $L_{CE}(\theta) = \sum_i \log p(x_i|\mathcal{N}(x_i); \theta)$. The neighborhood function $\mathcal{N}(x_i)$ must be highly tuned, and even slight variations in it can have significant impact on error. More crucially, the bias introduced by choosing $\mathcal{N}(x_i)$ is difficult to predict and unintuitive.

Haghighi and Klein (2006b) take prototypes as input to their method, and then uses SVD to link up words to prototypes with similar co-occurrence patterns (e.g. "Inner Richmond" has the label NEIGHBORHOOD). Haghighi and Klein (2006a) extends this framework to context-free grammar induction. Another group that has investigated integrating constraints into structure output learning is Chang et al. (2007) which integrates constraints into unsupervised learning with HMM. In their method, they reranking candidate labelings by constraint violations and then use a threshold set of these candidates for re-training in a Viterbi-like approximation to expectation maximization.

Generalized expectation criteria are unique in that it uses the expected distribution as the sole criterion for optimizing the model parameters on one set of unlabeled data (though it may also use labeled data). Most other methods do not try to directly fit these expectations, but use them instead as heuristics within a more complicated semi-supervised learning model. When we compare with techniques that use the label distribution (e.g. naïve Bayes with a fixed label prior), we find they do worse than GE, which demonstrates that GE is able to use these class distributions more effectively than other methods.

3. Label regularization is impossible under the Schapire et al. (2002) model, since if the model exactly matched the label expectation on a per-instance basis, in application it would assign all instances to the majority class.

3. Generalized Expectation Criteria

When a person is designing a classifier, they frequently have intuitions about the data they are trying to classify. For example, someone designing a part-of-speech tagger might know that 'nouns' are quite frequent, whereas 'conjunctions' are much less common. Manually labeling data can be a round-about way of providing this information to the machine learning model, and weak labeling provides another route for biasing the model with this information. It would be difficult to hypothesize priors over *model parameters* to capture this intuition. With GE the designer sets priors over *model expectations*, and since these expectations have a relatively transparent interpretation to the human designer, they provide an appealing route for injecting bias into the classifier. GE can effectively learn from a wide variety of side information, including expectation constraints which hold over global properties of the classifier (e.g. label marginals in label regularization), constraints over individual instances, and expectation constraints which are more expressive than the base model can model directly (e.g. a three label sequence in a markov order 1 CRF).

A **generalized expectation (GE) criterion** is a term in a parameter estimation objective function that assigns scores to values of a model expectation. First some standard notation: x is the input, y the output, and θ the parameters for a given model. Given a set of unlabeled data $\mathcal{U} = \{x\}$ and a conditional model $p(y|x;\theta)$, a GE criterion $G(\theta;\mathcal{U})$ is defined by a score function S and a constraint function $G(x,y)$:

$$G(\theta;\mathcal{U}) = S(E_{\mathcal{U}}[E_{p(y|x;\theta)}[G(x,y)]])\,.$$

In this light, GE criteria can be viewed as an replacement for a maximum likelihood estimator and can be maximized alone to yield a parameter estimate θ. For a particular choice of model family and parameterization, many different choices for score functions and constraint functions may be explored. In this paper we consider a subset of GE criteria which express a preference for a particular value of a constraint $\tilde{g}_{x,y}$, and apply the KL-divergence to compute model divergence from this constraint:

$$G(\theta;\mathcal{U}) = D(\tilde{g}_{x,y}||E_{\mathcal{U}}[p(y|x;\theta)G(x,y)])\,.$$

Other work has considered squared loss and constraint functions which are more and less expressive than the model parameterization (Druck et al., 2009a).

GE criteria can be used as a sole criterion for an objective function (e.g. Mann and McCallum (2008)). In this work, we combine it the log-likelihood, $L(\theta)$ to form a composite objective function:

$$O(\theta;\mathcal{U},\mathcal{D}) = L(\theta;\mathcal{D}) + G(\theta;\mathcal{U})\,.$$

Alternatively, an entropy regularization term (Grandvalet and Bengio, 2004)[4] can be combined into the above objective function in the same manner:

$$O(\theta;\mathcal{U},\mathcal{D}) = L(\theta;\mathcal{D}) + G(\theta;\mathcal{U}) + H(\theta;\mathcal{U})\,.$$

GE criteria can be interpreted as a generalization of traditional maximum likelihood. First, GE allows a variety of scoring functions (e.g. KL-divergence or mean-squared error from a reference

4. Entropy regularization cannot be framed as a instance of GE, but a generalization could encompass both: $G(\theta;\mathcal{U}) = S(E_{\mathcal{U}}[E_{p(y|x;\theta)}[F(x,y)]])$, where F is an arbitrary function over a particular (x,y) tuple (e.g. for entropy regularization $F = \log p(y|x;\theta)$.

distribution) and thus can incorporate information from a source other than empirical feature counts derived from the training data (e.g. human intuition, empirical counts derived from alternative data sources). Second, there need not be a one-to-one relationship between GE terms and features. For example, we can express preferences on a subset of model features (and leave others unconstrained), or on marginal distributions larger than model factors. In this paper, we apply a constraint for only one feature obtained from human intuition about the problem.

We explore **label regularization**, where the constraints $\tilde{g}$ are expectations of model marginal distributions on the expected output labels. We look at functions $G(x, y) = \mathbf{1}(y)$, and use various estimated label marginal distributions:

$$\tilde{g}_{x,y} = \tilde{p}(y)\,.$$

The effect of adding this term is to ensure that the model applied to the unlabeled data matches the label proportions.[5] Note that this does not force the conditional label distribution for each instance to conform to this constraint, but rather it encourages the model to meet this constraint in aggregate over all instances.

Similar to label regularization, Quadrianto et al. (2008) uses label proportions to learn classifiers, though there are some interesting differences from our work. Their method relies on having multiple training subsets with very different class distributions, whereas we only use one data set with a single set of label proportions. Their work concentrates on non-structured classification, whereas we extend our method to the structured output case.

Recent Work on GE Criteria and Related Methods

Since the initial proposal of GE criteria (under the name *expectation regularization*) in Mann and McCallum (2007), there has been a flurry of recent work on generalized expectation criteria and related methods which apply expectation constraints for weak learning.

In a set of user experiments, Druck et al. (2008) compares traditional labeled data and *labeled features* (which can be used to build feature marginal distributions). That study finds that given the same amount of time, human annotation in the form of labeled features and classifiers trained using GE criteria outperform human annotation of traditional labeled instances and maximum likelihood training. For the structured output case, Mann and McCallum (2008) has shown that expectations over features for CRF learning, similar to the prototypes proposed in Haghighi and Klein (2006b), are more effective when used with GE than similar numbers of labeled tokens used to train a CRF. Druck et al. (2009a) extends these methods to conditional random field dependency parsing models, and shows that in that case as well feature marginal distributions can be effectively used to guide training.

A few groups have investigated related methods for incorporating expectation constraints. Ganchev et al. (2009) use expectation constraints over aligned sentences and a source-language parser induce dependency grammar on a target language, using a generative method related to expectation-maximization and a discriminative model closely related to GE criteria. Bellare et al. (2009) presents an alternative objective function to learn using expectation constraints over unlabeled data.

Since the emphasis is on reducing human annotation time, it is a clear question as to whether active learning can be applied to help choose labeled features or expectation constraints. Druck

5. The mathematics is unchanged for expectation constraints over any single features, but the experiments concern only this simple scenario.

et al. (2009b) pursues this question and uses active learning to choose which features to use with GE criteria. Along those lines, Liang et al. (2009) proposes a notion of 'measurements' to encapsulate the variety of weakly labeled data, and uses active learning to guide which measurements are provided from the human annotator to guide learning.

4. GE Criteria for Log-linear Models

In this section, we describe how to apply the GE criteria proposed above to conditionally trained log-linear models, starting with conditional maximum-entropy models, aka multinomial logistic regression models, (Berger et al., 1996). In these models, there are k scalar feature functions $\psi_k(z, y)$, and the probability of the label y for input x is calculated by

$$p(y|x;\theta) = \frac{1}{Z(x)} \exp\left(\sum_k \theta_k \psi_k(x, y)\right),$$

where $Z(x) = \sum_{y'} \exp(\sum_k \theta_k \psi_k(x, y'))$ is the partition function. Given training data $\mathcal{D}$, the model is trained by maximizing the log-likelihood of the labels (with a Gaussian prior for regularization):

$$\begin{aligned} O(\theta;\mathcal{D}) &= \log L(\theta;\mathcal{D}) \\ &= \sum_{d\in\mathcal{D}} \log p(y_d|x_d;\theta) - \frac{\sum_k \theta_k^2}{2\sigma^2}\,. \end{aligned}$$

This can be done by gradient methods Malouf (2002), where the gradient of the likelihood is

$$\frac{\partial}{\partial\theta_k} O(\theta;\mathcal{D}) = \sum_d \left(\psi_k(x_d, y_d) - \sum_y p(y|x_d;\theta)\psi_k(x_d, y)\right) + \frac{\theta_k}{\sigma^2}.$$

For semi-supervised discriminative training, we augment the objective function by adding the generalized expectation criteria objective function terms.

$$\begin{aligned} O(\theta; D, U) &= L(\theta;\mathcal{D}) + G(\theta;\mathcal{U}) \\ &= \sum_d \log p(y_d|x_d);\theta) - \frac{\sum_k \theta_k^2}{2\sigma^2} - \lambda D(\tilde{g}_{x,y}||E_{\mathcal{U}}[E_{p(y|x;\theta)}[G(x, y)]])\,. \end{aligned}$$

Note that here the side information comes in the expectation constraints $\tilde{g}_{x,y}$ which specify particular priors for model marginal. In practice, we find that the hyper-parameters do not need to be extensively tuned. In particular, λ does not need tuning for each data set, and can be set simply to $\lambda = 10 \times$ # labeled examples.[6]

6. As support for this value of λ, notice that the KL-divergence is significantly smaller than the likelihood as the likelihood is proportional to the number of examples, while the KL-divergence is not.

The form of the GE criteria lends itself to optimization by gradient based methods. After dropping terms which are constant with respect to the partial derivative, we are left with:

$$\begin{aligned}\frac{\partial}{\partial\theta_k}G(\theta;\mathcal{U}) \propto & \frac{\partial}{\partial\theta_k}\sum_y \tilde{g}_{x,y}\log\sum_{x\in\mathcal{U}} p(y|x;\theta)G(x,y) \\ = & \sum_y\left(\frac{\tilde{g}_{x,y}}{\sum_{x\in\mathcal{U}}p(y|x;\theta)G(x,y)}\right)\sum_{x\in\mathcal{U}}\frac{\partial}{\partial\theta_k}p(y|x;\theta)G(x,y) \\ = & \sum_y\left(\frac{\tilde{g}_{x,y}}{\sum_{x\in\mathcal{U}}p(y|x;\theta)G(x,y)}\right)\sum_{x\in U}p(y|x;\theta)G(x,y)\left(\psi_k(x,y)-\sum_{y'}p(y'|x;\theta)\psi_k(x)\right).\end{aligned}$$

GE criteria aren't convex, and this can be shown by a contradictory example. Take a simple version of the GE criteria: $G(\theta;\mathcal{U}) = \sum_y \log\sum_x p(y|x;\theta)$. In this setting, for an arbitrary label y_i you can find parameter settings where $\forall x, p(y_i|x) = 1$, by having a parameter $\theta = \{\theta_k = \infty, \forall j \neq k, \theta_j = 0\}$ where $\psi_k(x,y) = \mathbf{1}\{y = y_i\}$. In this setting, it's pretty clear that multiple optima exist, and that other settings of θ yield smaller values of $G(\theta;\mathcal{U})$.

Label regularization can occasionally find a degenerate solution where, rather than the expectation of all instances matching the input distribution, instead, the distribution over labels for *each instance* will match the given distribution on every example. For example, given a three class classification task, if the labeled class distribution $\hat{p}_j(y) = \{.5, .35, .15\}$, it will find a solution such that $\tilde{p}(y;\theta) = \{.5, .35, .15\}$ for every instance. As a result, all the test instances will be assigned the same label.

One solution, appealing to 0/1 loss, would be to simply measure and match the expectation over winning class counts, but this is not differentiable. So instead, we make $p(y|x;\theta)$ more peaked using a less than 1.

$$p(y|x;\theta) \propto \exp\left(\frac{1}{T}\sum_k \theta_k\psi_k(x)\right).$$

This is differentiable and thus amenable to many gradient ascent methods. In practice we find that this meta-parameter does not require fine-tuning. Across all data sets we simply use $T = 0.1$ for multi-class problems and $T = 1$ for binary classification problems, and we find this to work well.

4.1 CRF Training

The previous section has shown the application of generalized expectation criteria to classification models. However, GE can additionally be applied to structured models. In this section, we examine the case of linear chain structured conditional random fields (Lafferty et al., 2001), and derive the GE gradient for this model.

Linear-chain CRFs are a discriminative probabilistic model over sequences $\mathbf{x} = \langle x_1..x_n\rangle$ of feature vectors and label sequences $\mathbf{y} = \langle y_1..y_n\rangle$, where $|\mathbf{x}| = |\mathbf{y}| = n$, and each label $y_i \in s$. This model is analogous to maximum entropy models for structured outputs, where expectations can be efficiently calculated by dynamic programming. For a linear-chain CRF of Markov order one

$$p(\mathbf{y}|\mathbf{x};\theta) = \frac{1}{Z(\mathbf{x})}\exp\left(\sum_k \theta_k\Psi_k(\mathbf{x},\mathbf{y})\right),$$

where $\Psi_k(\mathbf{x}, \mathbf{y}) = \sum_i \psi_k(\mathbf{x}, y_i, y_{i+1}, i)$, and the partition function $Z(\mathbf{x}) = \sum_{\mathbf{y}'} \exp(\sum_k \theta_k \Psi_k(\mathbf{x}, \mathbf{y}'))$. Given training data $\mathcal{D}$, the model is trained by maximizing the log-likelihood,

$$O(\theta; \mathcal{D}) = \sum_d \log p(\mathbf{y}_d | \mathbf{x}_d; \theta) - \frac{\sum_k \theta_k^2}{2\sigma^2},$$

by gradient-based methods where the gradient of the likelihood is (similar to the non-structured case):

$$\frac{\partial}{\partial \theta_k} \mathcal{O}(\theta; D) = \sum_d \left(\Psi_k(\mathbf{x}_d, \mathbf{y}_d) - \sum_{\mathbf{y}} p(\mathbf{y} | \mathbf{x}_d; \theta) \Psi_k(\mathbf{x}_d, \mathbf{y}) \right) + \frac{\theta_k}{\sigma^2}.$$

The second term (the expected counts of the features given the model) can be computed in a tractable amount of time, since according to the Markov assumption, the feature expectations can be rewritten:

$$\sum_{\mathbf{y}} p(\mathbf{y} | \mathbf{x}; \theta) \Psi_k(\mathbf{x}, \mathbf{y}) = \sum_i \sum_{y_i, y_{i+1}} p(y_i, y_{i+1} | \mathbf{x}; \theta) \psi_k(\mathbf{x}, y_i, y_{i+1}, i).$$

A dynamic program (the forward/backward algorithm) then computes in time $O(n|s|^2)$ all the needed probabilities $p_\theta(y_i, y_{i+1})$, where n is the sequence length, and $|s|$ is the number of labels.

4.2 Semi-supervised training with Generalized Expectation Criteria

To add unlabeled data regularization to the CRF training, just as with the maximum entropy model, we augment the objective function with the regularization term:

$$\begin{aligned} O(\theta; \mathcal{D}, \mathcal{U}) =& L(\theta; \mathcal{D}) + G(\theta; \mathcal{U}) \\ =& \sum_d \log p(\mathbf{y}_d | \mathbf{x}_d; \theta) - \frac{\sum_k \theta_k^2}{2\sigma^2} - \lambda D(\tilde{g}_{x,y} || E_{\mathcal{U}}[E_{p(\mathbf{y}|\mathbf{x};\theta)}[\sum_i G(\mathbf{x}, y_i)]]). \end{aligned}$$

Note that we restrict the constraint function to functions over one output label, $G(\mathbf{x}, y_i)$. Druck et al. (2009a) has looked at extending the method to arbitrary constraint functions $G(\mathbf{x}, \mathbf{y})$, but here we only consider constraint functions over functions of one label.

The derivation of the gradient for $G(\theta; \mathcal{U})$ is somewhat more complicated than in the unstructured case, but follows roughly the same line. s is the set of permissible output labels. $\mathbf{y}_{(m=s)} = \{\mathbf{y} : y_m = s\}$. The gradient is then:

$$\frac{\partial}{\partial \theta_k} G(\theta; \mathcal{U}) \propto \frac{\partial}{\partial \theta_k} \sum_s \tilde{g}_{\mathbf{x},s} \log \sum_{\mathbf{x} \in \mathcal{U}} \sum_m \sum_{\mathbf{y}_{(m=s)}} p(\mathbf{y}_{(m=s)}|\mathbf{x}) G(\mathbf{x}, s)$$

$$= \sum_s \left(\frac{\tilde{g}_{\mathbf{x},s}}{\sum_{\mathbf{x} \in \mathcal{U}, m, \mathbf{y}_{(m=s)}} p(\mathbf{y}_{(m=s)}|\mathbf{x}) G(\mathbf{x}, s)} \right) \sum_{\mathbf{x} \in \mathcal{U}} \sum_m \sum_{\mathbf{y}_{(m=s)}} \frac{\partial}{\partial \theta_k} p(\mathbf{y}_{(m=s)}|\mathbf{x}) G(\mathbf{x}, s)$$

Now define: $\frac{\tilde{g}}{G} = \left(\frac{\tilde{g}_{\mathbf{x},s}}{\sum_{\mathbf{x} \in \mathcal{U}, m, \mathbf{y}_{(m=s)}} p(\mathbf{y}_{(m=s)}|\mathbf{x}) G(\mathbf{x}, s)} \right)$

$$= \sum_s \frac{\tilde{g}}{G} \sum_{\mathbf{x} \in \mathcal{U}} \sum_m \sum_{\mathbf{y}_{(m=s)}} p(\mathbf{y}_{(m=s)}|\mathbf{x}) G(\mathbf{x}, s) \left(\Psi_k(\mathbf{x}, \mathbf{y}_{(m=s)}) - \sum_{\mathbf{y}'} p(\mathbf{y}'|\mathbf{x}) \Psi_k(\mathbf{x}, \mathbf{y}) \right)$$

$$= \sum_s \frac{\tilde{g}}{G} \sum_{\mathbf{x} \in \mathcal{U}} \sum_m \sum_{\mathbf{y}_{(m=s)}} p(\mathbf{y}_{(m=s)}|\mathbf{x}) G(x, s) \Psi_k(\mathbf{x}, \mathbf{y}_{(m=s)})$$

$$- \sum_s \frac{\tilde{g}}{G} \sum_{\mathbf{x} \in \mathcal{U}} \sum_m \sum_{\mathbf{y}_{(m=s)}} p(\mathbf{y}_{(m=s)}|\mathbf{x}) G(\mathbf{x}, s) \sum_{\mathbf{y}'} p(\mathbf{y}'|\mathbf{x}) \Psi_k(\mathbf{x}, \mathbf{y}) \Bigg)$$

$$= \sum_s \frac{\tilde{g}}{G} \sum_{\mathbf{x} \in \mathcal{U}} \left(\sum_i \sum_{y_i, y_{i+1}} \psi_k(\mathbf{x}, y_i, y_{i+1}, i) \sum_m p(y_i, y_{i+l}, y_m = s|\mathbf{x}) G(\mathbf{x}, s) \right)$$

$$- \sum_s \frac{\tilde{g}}{G} \sum_{\mathbf{x} \in \mathcal{U}} \left(\sum_i \sum_{y_i, y_{i+1}} \psi_k(\mathbf{x}, y_i, y_{i+1}, i) \right) \left(\sum_m p(y_m = s|\mathbf{x}) G(\mathbf{x}, s) \right).$$

After combining terms and rearranging we arrive at the final form of the gradient:

$$= \sum_{\mathbf{x} \in \mathbf{U}} \sum_i \sum_{y_i, y_{i+1}} \psi_k(\mathbf{x}, y_i, y_{i+1}, i) \times$$

$$\sum_s \frac{\tilde{g}}{G} \left(\sum_m p(y_i, y_{i+1}, y_m = s|\mathbf{x}; \theta) G(\mathbf{x}, s) - p(y_i, y_{i+1}|\mathbf{x}; \theta) \sum_m p(y_m = s|\mathbf{x}; \theta) G(\mathbf{x}, s) \right).$$

Here, the second term is easily obtainable from forward/backward, but the first term is a little more complicated to compute. Computing this term naively would require multiple runs of constrained forward/backward. Here we propose a more efficient method that requires only one run of forward/backward.[7] For the sake of simplicity, we omit the constraint function $G(\mathbf{x}, s)$; its addition is trivial. First we decompose the probability into two parts:

$$\sum_m p(y_i, y_{i+1}, y_m = s|\mathbf{x}; \theta) = \sum_{m=1}^{i} p(y_i, y_{i+1}, y_m = s|\mathbf{x}; \theta) + \sum_{m=i+1}^{n} p(y_i, y_{i+1}, y_m = s|\mathbf{x}; \theta).$$

7. Kakade et al. (2002) present a related method that computes $p(y_{1..i} = s_{1..i}|y_{i+1} = s)$.

Similar to forward/backward we build a lattice of intermediate results that then can be used to calculate the quantity of interest:

$$\begin{aligned}
&\sum_{m=1}^{i} p(y_i, y_{i+1}, y_m = s|\mathbf{x};\theta) \\
&= p(y_i, y_{i+1}|\mathbf{x};\theta)\delta(y_i, s) + \sum_{m=1}^{i-1} p(y_i, y_{i+1}, y_m = s|\mathbf{x};\theta) \\
&= p(y_i, y_{i+1}|\mathbf{x};\theta)\delta(y_i, s) + \left(\sum_{y_{i-1}}\sum_{m=1}^{i-1} p(y_{i-1}, y_i, y_m = s|\mathbf{x};\theta)\right) p(y_{i+1}|y_i, \mathbf{x};\theta)\,.
\end{aligned}$$

For each label s, it requires one pass to create a lattice with $\sum_{m=1}^{i-1} p(y_{i-1}, y_i, y_m = s|\mathbf{x};\theta)$ for all pairs (y_i, y_{i+1}). $\sum_{m=i+1}^{n} p(y_{i-1}, y_i, y_m = s|\mathbf{x};\theta)$ can be computed in the same fashion. To compute the lattices it takes time $O(n|s|^2)$, and one lattice must be computed for each label so the total time is $O(n|s|^3)$.

5. Experimental Results for Classifiers

We present two sets of experiments: experiments on maximum entropy models and conditional random fields for the special case of generalized expectation criteria, *label regularization*. For this set of experiments, we evaluate on five different data sets, and compare against seven different semi-supervised and supervised-only methods. We present learning curves, where the amount of labeled training data is gradually increased from one instance per class up to thousands of instances and demonstrate that generalized expectation criteria are able to show improvements for both types of scenarios. We present experiments with noisy expected distributions, and show that the method is robust with respect to a variety of settings for λ and temperature. We do not vary the gaussian regularizer, but leave a default value. Unpublished experiments by G. Druck[8] have suggested that while gaussian regularization can have an effect for label regularization, for more complicated GE variants it doesn't dramatically affect performance. We begin training with parameters set at 0 (even though the objective function may not be convex).

5.1 Experimental Set-up

First, we examine a protein secondary structure prediction task (**SecStr**), as extensively evaluated in Chapelle et al. (2006), compare with the published results and show that label regularization is able to outperform previous methods. Next, we examine three especially difficult natural language processing tasks: the CoNLL03 named-entity recognition task (**CoNLL03**), Part of speech tagging of the Wall Street Journal (**POS**), and the 2007 BiocreativeII evaluation (**BIOII**), using a sliding window classifier[9]. Finally, one of the main targets for semi-supervised learning is text classification (Nigam et al., 2006), and we evaluate on the simulated/real auto/aviation (**SRAA**) task. The tasks are large in scale, with up to hundreds of thousands of instances and features (see Table 1).

8. Personal communication.

9. The sliding window classifier makes independent decisions for each element in the sequence. While finite-state methods could also be applied in these cases, the cost of training label regularization would be prohibitive, and we found that these methods work well.

Name	# Test Examples	# Unlabeled Examples	# features	# classes
SRAA	20k	20k	77,494	4
POS	20k	20k	11,520	44
SecStr	1000	83k	314 (45,436)	2
BIOII	100k	100k	54,958	3
CoNLL03	100k	100k	114,264	9

Table 1: The data sets are complex: they have dramatic class skews, highly inter-dependent features, and large amounts of data. The SecStr data set has 315 atomic features, and 45k features when pairwise feature conjunctions are used.

They have complex characteristics such as heavily inter-dependent features and highly skewed class distributions.

Across all of the experiments, for supervised comparisons, we compare with naïve Bayes and maximum entropy models, for semi-supervised comparisons we compare with naïve Bayes trained with EM and maximum entropy models trained with entropy regularization. On some tasks, in particular the sliding window NLP tasks, the number of features per instance varied dramatically, and so we used document length normalization for the naïve Bayes approaches as we found it to significantly improve accuracy. On the secondary structure prediction (**SecStr**), we had access to published results for a supervised SVM using a radial-basis function (RBF) kernel, a Cluster Kernel (Weston et al., 2006) and a graph based-method, the Quadratic Cost Criterion with Class Mean Normalization (Bengio et al., 2006) trained using various data sub-sampling schemes (Delalleau et al., 2006): a random sampler and two smarter variations. Presumably, training a graph-based method on the entire unlabeled training set would have been technically infeasible.

For **CoNLL03**, **POS**, **BIOII**, and **SRAA**, we performed inductive learning, splitting the data randomly into two sections, training and test. From the training set, we randomly chose some instances to be labeled and set the remainder to be hidden. Out of those hidden, we then select a fixed number to use for unsupervised learning. We then evaluate the model on the hidden test data. We repeat this evaluation five times for each of the models.

The **SecStr** task was set up in what is commonly called transductive learning, where the model is evaluated on hidden training data. For this task, the labeled/unlabeled splits were provided with the data from Chapelle et al. (2006) and evaluation is on hidden training data. In order to provide a somewhat more fair comparison with the RBF kernels used by the other methods on this task, the feature set used by the maximum entropy model and naïve Bayes models is augmented by pairwise feature conjunctions.[10]

For the maximum entropy model trained with entropy regularization, after some experimentation, we weighted its contribution to the objective function with

$$\lambda = \text{\# labeled data points / \# unlabeled data points}\,.$$

This was shown to yield relatively good performance. For the first set of experiments, we use label proportions estimated from all of the data, corresponding to a use-case where a user gives this

10. Though, as one anonymous reviewer noted, this makes them strictly more expressive then kernel methods with quadratic features.

	# Labeled Instances		
	2	100	1000
SVM (supervised)		55.41	**66.29**
Cluster Kernel		57.05	65.97
QC randsub (CMN)		57.68	59.16
QC smartonly (CMN)		57.86	59.29
QC smartsub (CMN)		57.74	59.16
Naive Bayes (supervised)	52.42% (± 0.4)	57.12% (± 0.7)	64.47% (± 0.2)
Naive Bayes EM	50.79% (± 1.5)	57.34% (± 1.1)	57.60% (± .009)
MaxEnt (supervised)	52.42% (± 0.5)	56.74% (± 1.1)	65.43% (± 0.3)
MaxEnt + Ent. Min.	49.40% (± 2.1)	54.45% (± 1.8)	58.28% (± 0.1)
MaxEnt + GE	**57.08%** (± .03)	**58.51%** (± 0.4)	65.44% (± 0.3)

Table 2: Label regularization outperforms other semi-supervised learning methods at 100 labeled data points. At one instance per class, its performance is better than the *supervised* SVM and maximum entropy model at 100. Standard error is reported for experiments that were run locally. Other experimental performance is taken from the literature.

knowledge to the system during training. Section 5.3 presents experiments showing robustness to noisy label proportions both when smoothed towards a uniform distribution and when sampled from a limited number of training examples. Across the experiments, we observed that label regularization trains in time linear in the amount of unlabeled data, and since the resulting model is parametric, it is linear in the number of features for evaluation.

5.2 Learning Curves

For the first set of experiments, we experimented with varying the number of supervised training example, while keeping the unlabeled data set size the same (with sizes of the unlabeled data shown in Table 1). We added examples in a balanced way, going from 1 example per class up to 500 examples per class (when possible). For each data point we ran 5 trials, where the data was partitioned into training/test/unlabeled uniformly at random. The sole exception was for the **SecStr** data where the data splits and tests were pre-specified.

As Table 2 shows, for **SecStr**, label regularization outperforms the other methods at 100 labeled points, and approaches the cluster kernel method on 1000 points. While performance results were not available for the other methods for two instances per class, we ran label regularization for this case and found that it outperforms the *supervised* SVM and maximum entropy model when they are trained with 100 labeled points. In these experiments QC is not run over the complete data (presumably because of scalability problems), but operates on a subset, either selected randomly (randsub) or in a smarter fashion (smartonly and smartsub), while the label regularization method uses the complete data.

Figures 1, 2, 3, and 4 show classifier performance using a fixed amount of unlabeled data as greater amounts of labeled data is added. Label regularization yields significant benefits over the other methods for **POS**, **BIOII**, and **CoNLL03** for all amounts of labeled data. Label regularization on **SRAA** shows a benefit over the fully supervised maximum entropy model but its accuracy is

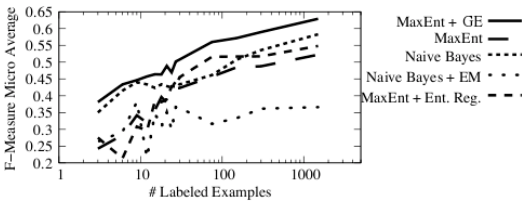

Figure 1: **BIOII**: Label regularization (GE) outperforms all other methods. The x-axis represents increasing numbers of labeled data instances. The y-axis is the F-measure micro average across all classes.

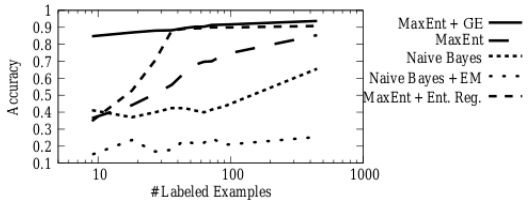

Figure 2: **CoNLL03**: Label regularization (GE) outperforms all other methods. The x-axis represents increasing numbers of labeled instances per class, and the y-axis is accuracy.

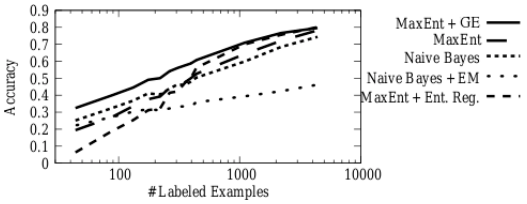

Figure 3: **POS**: Label regularization (GE) outperforms all other methods, though performance improvements over supervised maximum entropy methods appear to level off at 1300 labeled instances.

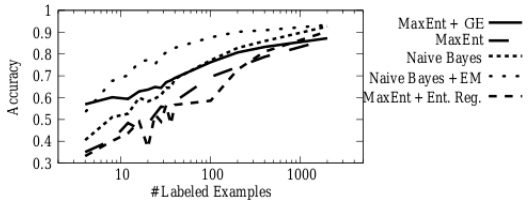

Figure 4: **SRAA**: Label regularization (GE) outperforms its supervised maximum entropy counterpart and entropy regularization and is the winner at one labeled instance per lass. After that, naïve Bayes EM is the clear winner.

not as high as that obtained by the EM-trained naïve Bayes learner. This may be partly explained by the fact that the baseline performance of the discriminative maximum entropy model is much lower than the generative naïve Bayes model, so that label regularization starts off at a considerable deficit.

While alternative methods often result in degredations of performance over their supervised counter parts (EM, entropy regularization, cluster kernels), in these experiments label regularization consistently yielded improved accuracy. Additionally, the benefit of label regularization is more apparent as the feature sets and numbers of unlabeled instances increase, with the least improvements on one of the simplest tasks, the **SRAA** text classification task.

These experiments demonstrate that label regularization can at least match, and in many cases beat, alternative methods of semi-supervised learning, given minimal additional information and access to large samples of unlabeled data. The successes of GE suggest more investigation of additional, alternative modalities of supervision which will generate data that can be added to supervised classifiers and combined with unlabeled data in order to improve performance.

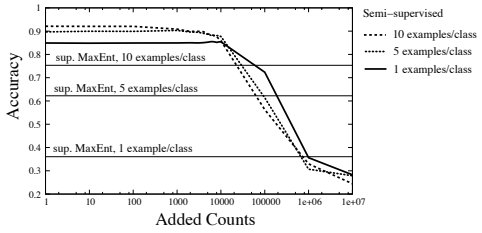

Figure 5: **CoNLL03**: The x-axis represents increasing amount of noise towards a uniform distribution. On this data set, the majority class is 84% of the instances, and so the uniform distribution is an extremely poor approximation. Performance suffers little when the majority class proportion is erroneously given as 61%($\nu = 10,000$)

5.3 Noisy Priors

The previous section assumes that the system has accurate knowledge of the distributions over the labels. In this section, we perform a sensitively analysis by gradually smoothing the class distribution until it reaches a uniform distribution. We add noisy counts ν to the true counts $c(y)$:

$$\hat{p}_j(\mathbf{y})(y) = \frac{c(y) + \nu}{\sum_{y'} (c(y') + \nu)} .$$

As more noise is added, the input distribution converges to uniform.

Figure 5 demonstrates the effect of increasing noise in the system. At $\nu = 1,000$, the majority class probability drops from 84% to 80% and there is almost no loss of performance. At $\nu = 10,000$ are added, the majority class probability drops to 61% and there is only a slight loss of performance. At $\nu = 1e07$ the majority class probability has dropped to 11%, a virtually uniform distribution, and performance has leveled off. These results are encouraging as they suggest that relatively large changes (of 20% absolute, 27% relative) can be tolerated without major losses in accuracy. Even when the human has no domain knowledge to contribute, label distribution estimates of sufficient accuracy should be obtainable from a reasonably small number of labeled examples.

To test this assumption, we performed another set of experiments, where instead of smoothing the input distributions towards a uniform distribution, we sampled them from the data, varying the number of instances used in the sample. These points were sampled from the data, and then the data was partitioned into test/train/unlabel splits. Figure 6 and 7 demonstrate the effect of sampled distributions, as opposed to distributions smoothed towards a uniform distribution. For the **CoNLL3** data set, as can be seen in Figure 6, after sampling from 1000 points, the performance of the classifier doesn't get worse, suggesting that only a small amount of prior knowledge or labeling is necessary for determining accurate input distributions. With the **POS** data, it appears that as you increase the number of points used to compute the sample performance improves, though it

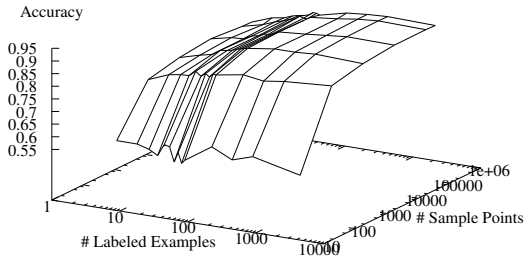

Figure 6: **CoNLL03**: An input label distribution with sampling noise. After 1000 points, sampling from more points doesn't appear to lead to performance improvements.

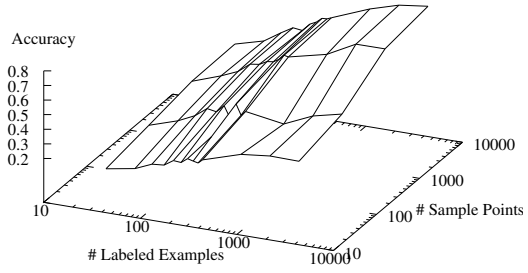

Figure 7: **POS**: An input label distribution with sampling noise. Since POS has many classes (44), access to an accurate sampling distribution has a larger impact on performance, and the graph suggests that even higher precision in sampling would lead to higher accuracy. Note that we were unable to sample as many points as in the previous example beause of limited amounts of available data.

appears to begin to level out at 1000 points. Because this data set is significantly smaller, we were unable to continue running experiments with larger numbers of sampled points to evaluate when the performance begins to level out.

5.4 Robustness

Along with robustness in the face of noise from the estimated label proportions, the model is robust to changes in λ and temperature. As can be seen in Figure 8, λ and temperature have a wide plateau over which their performance is stable. At some extreme values of λ and temperature, the performance degrades, and can drop below supervised performance. This trend was observed for 500 labeled examples (shown in the figure), as well as in cases when there as little as one labeled example for a number of the data sets. For other semi-supervised techniques such as entropy regularization, extensive tuning is required across for each individual data set and labeled/unlabeled data set sizes in order to improve upon supervised-only performance (Jiao et al., 2006).

5.5 Running Time

Figure 9 shows the running time per optimization iteration for the two largest tasks, **CoNLL** and **BIOII**. The slope variation between the running times can be accounted for by the number of features in each of the data sets.

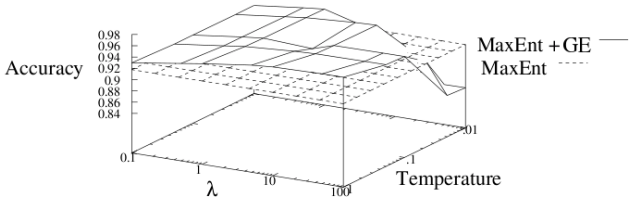

Figure 8: **CoNLL03**: For a wide range of λ and temperature the performance is similar and surpasses the purely supervised performance.

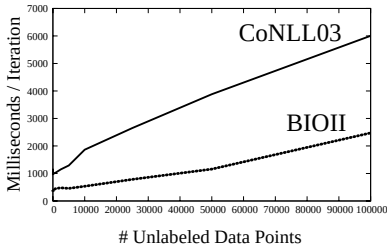

Figure 9: Label regularization is linear in the number of unlabeled examples, without requiring sub-sampling.

5.6 Mechanism of Effect

One question that needs to be addressed is whether label regularization is improving performance solely by adjusting the label proportions or operating in some other fashion. Though certainly correcting label proportions is one pathway for improved performance we have two pieces of evidence that additional learning is happening in the model. First, when we allow other classifiers access to the label proportions, they are unable to reach the same gains as achieved with GE. For example, in all of the experiments the naïve Bayes classifier has its label prior fixed to the input distribution as it is subsequently trained with EM, yet it typically fails to reach the same level of performance as achieved with the GE methods. Second, Druck et al. (2008) reports results of experiments using GE criteria where the input distributions were only allowed to affect the features they were conditioned on (corresponding to only being able to adjust the label proportions). In these experiments, when GE could only adjust the features specified by the input distributions it achieved significantly worse results, thus indicating that learning parameter values for features not directly conditioned by the input distributions has a dramatic effect on classifier accuracy.

5.7 Combining Label Regularization and Entropy Regularization

Generalized expectation criteria can often be easily combined with other models. Any semi-supervised model in the parametric model family, such as expected gradient methods (Salakhutdinov et al., 2003), can be easily combined with GE, and certain generative models such as naïve MRFs (Druck et al., 2007) can be simply combined as well. More distantly, just as various models can be augmented with regularization terms (as in ridge regression for linear regression models), GE may be augmented in the same way. In this paper we used a Gaussian prior and minimized KL-divergence

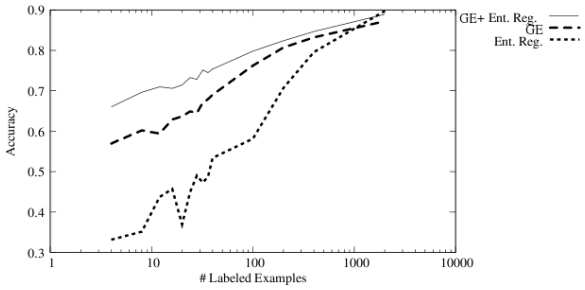

Figure 10: **SRAA:** Combining label regularization and entropy regularization can be easily accomplished, and yields improvements over label regularization alone for this data set.

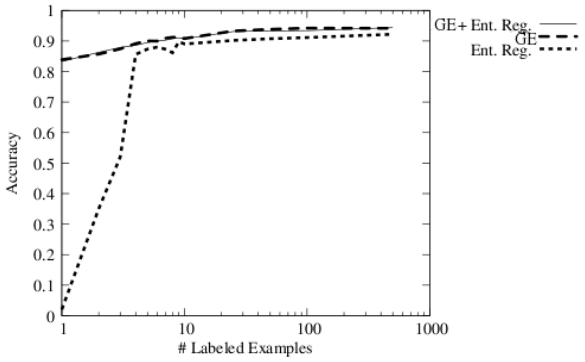

Figure 11: **CoNLL03:** On this data set, combining label regularization and entropy regularization does not lead to any benefit. The GE+Entropy regularization curve exactly overlaps the GE curve.

from input distributions with gradient methods here, in other cases it might require an alternative penalty term from the input distributions and a different minimization technique.

Here we examine combining label regularization with entropy regularization where the objective function is augmented with more than one regularization criterion. For many of the experiments, combining label regularization and entropy regularization does not lead to improvements. Two exceptions were experiments on **SRAA** and the **SecStr** data sets. Notably, on **SecStr**, combined entropy regularization and label regularization yields a performance of 66.30, a level which matches the performance of the supervised radial-basis SVM and beats all other unsupervised methods. For **SRAA**, Figure 10 shows that when entropy regularization is added to label regularization, there can sometimes be a benefit over the use of label regularization or entropy regularization alone. In comparison, Figure 11 shows that there is little or no difference in performance when entropy regularization is combined with label regularization in the case of **CoNLL03**.

MRF (prototypes)	53.7%
MRF (prototypes) + cluster	71.5%
HMM supervised only (100)	74.4%
CRF supervised only (100)	75.8% ($\pm$ 0.3)
CRF supervised (100) + Ent. Reg. (2k)	76.7% ($\pm$ 0.4)
CRF supervised (100) + GE (2k)	77.1% ($\pm$ 0.3)

Table 3: **APT**: A CRF trained semi-supervised with 100 labeled examples and 2k unlabeled exampled has the highest performance, beating the strictly supervised CRF and the CRF trained semi-supervised with entropy regularization. Standard error is shown in parentheses for experiments run locally.

6. Experimental Results for Conditional Random Fields

In this section, we examine the performance of label regularization for conditional random fields. We look at two data sets, Craigslist Apartment listings, and citation data. We compare against two previous methods for semi-supervised learning of conditional random fields, entropy regularization and the clustering method proposed by Miller et al. (2004) in Section 6.3. We demonstrate that label regularization can achieve higher accuracy than purely supervised training, and can beat or match the performance of entropy regularization or clustering. We do not vary the gaussian regularizer, but leave a default value. We begin training with parameters set at 0 (even though the objective function may not be convex). Later experiments start from a model initialized by a non-structured classifier trained with GE, which we observed to yield higher accuracy (Mann and McCallum, 2008).

6.1 Apartment Listings

First we examine performance on the apartment data set (**APT**) initially presented by Grenager et al. (2005) and later examined by Haghighi and Klein (2006b), with label regularization. This data was collected in June 2004 from craigslist.com, and consists of 302 hand-labeled ads where each ad is labeled with 12 fields (e.g. SIZE, RENT, NEIGHBORHOOD, FEATURES). The average ad has 119 tokens in 8.7 fields.

For this task, sliding window models perform poorly as the fields are "sticky," (i.e. the best way to predict the next label is from the previous label). We set label regularization λ as before, but set the entropy regularization λ to 0.01 times the number of labeled examples divided by the number of unlabeled examples. For these experiments we used 2,000 unlabeled apartments listings.

We performed minimal feature engineering, using only standard capitalization and word class features (e.g. "digits"). We additionally used a feature that is the exact token string of the previous word. The use of flexible, non-independent features demonstrates the benefit of the greater expressive power of discriminatively trained CRFs; with these features alone, the CRF out-performs the supervised HMM.

Table 3 compares the performance of our system relative to previous supervised and semi-supervised systems[11], where at the maximum amount of training data, label regularization achieves

11. We did not implement the ad-hoc boundary pre-processing and post-processing as is performed by the unsupervised MRF. These boundary features gave a 3% improvement for Haghighi and Klein (2006b).

# Supervised	Supervised	Entropy Regularization	GE: Estimated Priors	GE: Accurate Priors
1	41.2% (± 1.7)	44.3% (± 0.3)	41.3% (± 1.7)	**45.7%** (± 1.7)
5	48.4% (± 1.8)	45.8% (± 1.1)	51.9% (± 1.6)	**56.2%** (± 0.6)
10	55.9% (± 1.3)	58.0% (± 2.5)	57.5% (± 1.2)	**63.0 %** (± 0.2)
50	71.7% (± 0.5)	**74.1 %** (± 0.2)	73.7 % (± 0.4)	74.0 % (± 0.6)
100	75.8% (± 0.3)	76.7 % (± 0.4)	**77.4 %** (± 0.2)	77.1 % (± 0.3)

Table 4: **APT**: Use of input distributions estimated from limited labeled data yields smaller improvements over the true distributions, but still improves over the strictly supervised accuracy, and at large levels of training data, nearly matches the accuracy achieved by the true distributions. Standard error is shown in parentheses for the experiments.

a 1.3% accuracy improvement over the purely supervised CRF, and a 0.4% accuracy improvement over semi-supervised learning via entropy regularization.

Table 4 shows learning curve performance for the CRF with a variety of different settings. With accurate input distributions, the CRF trained with label regularization achieves the highest performance, for all but with one example, where entropy regularization is the highest performer. Label regularization gives a 8% absolute accuracy improvement at the lower levels of training data and a 1.3% boost (a 5% relative error reduction) for a highly trained sequence model. Unlike entropy regularization, label regularization improves over supervised learning across all amounts of training data.

In addition to testing with accurate proportions, we also examined performance when input distributions were read directly off of the minimal training label sequence. When these noisy input distributions were used, the performance was less than with entropy regularization at lower levels of training data, but it still provided consistent gains in performance across all training settings.

6.2 Citation Data

In addition to experiments on apartment data, we also ran experiments on citation data (**CITE**) as given by Grenager et al. (2005). This data set consists of 500 hand-annotated citations taken from the reference sections of different computer science papers. Each citation is annotated with 13 fields (e.g. AUTHOR, TITLE, DATE, JOURNAL), and on average the citation has 35 tokens annotated with 5.5 fields. For this experiment we used 1,000 unlabeled examples, with $\lambda = 1$ times the number of unlabeled data points, and the entropy regularizer set as before. On this set of data, as shown in Table 5, label regularization gives a slight win at low levels of training data, but at higher levels, it does not provide any improvement. Entropy regularization unfortunately consistently decreases performance. Here, estimating the input distributions from the minimal training data leads to performance decreases beyond supervised training. The difference between the two data sets suggest that more work is needed to understand in what situations label regularization can be expected to work well.

One thing to note from the experiments, is that as training data increases, the performance gap between the true distributions and estimated input distributions diminishes, until at 100 supervised examples, it almost matches the accuracy achieved by the true distributions. These results are at once encouraging and surprising, and suggest two possible reasons for improvement. First, perhaps

# Supervised	Supervised	Entropy Regularization	GE: Estimated Priors	GE: Accurate Priors
1	24.9% (± 3.6)	17.5% (± 3.2)	25.5% (± 3.8)	**37.3%** (± 2.1)
5	52.4% (± 0.5)	27.0% (± 2.7)	52.4% (± 5.6)	**54.7%** (± 0.5)
10	56.1% (± 0.6)	35.2% (± 3.3)	55.5% (± 0.7)	**57.9%** (± 0.4)
50	67.8% (± 0.6)	66.5% (± 0.8)	67.5% (± 0.3)	**68.0%** (± 0.6)
100	72.7% (± 0.5)	**73.5%** (± 0.5)	72.4% (± 0.3)	72.0% (± 0.6)

Table 5: **CITE**: Label regularization (GE) is able to significantly improve on the purely supervised cases at very low levels of training data. At higher levels of training data, it makes less of an impact. Standard error is shown in parentheses for the experiments.

# Supervised	Supervised	Supervised + Clustering	GE	GE + Clustering
1	41.2% (± 1.7)	44.5% (± 0.9)	45.7% (± 1.7)	**50.0%** (± 0.3)
5	48.4% (± 1.8)	54.3% (± 1.8)	56.2% (± 0.6)	**58.9%** (± 1.1)
10	55.9% (± 1.3)	61.2% (± 1.5)	63.0 % (± 0.2)	**64.4%** (± 0.9)
50	71.7% (± 0.5)	**74.1%** (± 0.5)	74.0 % (± 0.6)	**74.1%** (± 0.3)
100	75.8% (± 0.3)	**77.6%** (± 0.1)	77.1 % (± 0.3)	**77.6%** (± 0.2)

Table 6: **APT**: Using clustering features gives an additional gain to label regularization for low levels of training data, but it doesn't provide any additional benefit at higher levels of training data. Standard error is shown in parentheses for the experiments.

the CRF isn't matching the proportions implicit in the labeled data, even though it has access to this information. Second, it suggests that the unlabeled data does have an effect beyond that of helping to readjust the classifier towards the input distributions. In particular, perhaps new features are being brought in by inclusion of the unlabeled examples.

The strengths of label regularization over entropy regularization are two-fold. First, GE gives an overall win in performance across many different levels of training data. Second, GE gives consistent gains. GE rarely performs worse than purely supervised training (when the input distributions are accurate), whereas the performance of entropy regularization is erratic, sometimes yielding a gain in performance, in other cases leading to a severe decrease in performance.

6.3 Clustering Features

Miller et al. (2004) proposes a method of using unsupervised clusters to improve performance. In his method, the unlabeled data undergoes word clustering, and then features corresponding to clusters are added during supervised training. A similar method can be applied here, in which features corresponding to unsupervised word clusters are added during semi-supervised training. We applied this method to the apartment data, and for very low level of training, found encouraging performance gains. Table 6 shows that for one labeled example, 2,000 unlabeled examples, and unsupervised clustering, the system achieves almost a 5% improvement by using these unsupervised cluster features (45.7% to 50.0%). At higher levels of training data, label regularization and clustering features interact poorly, and using them together lead to no improvement. While at lower

levels of training data, label regularization is able to achieve more gains than the clustering method, at higher levels of performance, it only matches performance. We have also attempted to use sim features as proposed in Haghighi and Klein (2006b), but found these methods difficult to tune (e.g. what SVD rank to retain).

7. Conclusion

This paper has presented *generalized expectation criteria*, a simple, robust, scalable method for semi-supervised learning. This method penalizes models by divergence between the model's expectations over the unlabeled data and input conditional probabilities, which can be estimated from labeled data or given as a priori knowledge by a human annotator. An important special case, *label regularization* is empirically explored for the case of maximum entropy models where we find it to provide accuracy improvements over entropy regularization, naïve Bayes EM, Quadratic Cost Criterion (a representative graph-based method) and a cluster kernel SVM. We show that the method is robust to noise in the estimates of the input conditional probabilities, the meta-parameters need little or no tuning, and that it runs in linear time with increasing numbers of unlabeled examples.

We additionally present extensions of the method to conditional random fields. In this case, the gradient computation is complicated and requires the use of a specialized dynamic program which computes $\sum_j p(y_i, y_{i+1}, y_j = l|\mathbf{x}; \theta)$. Conditional random fields trained with label regularization outperform alternative methods for semi-supervised training such as entropy regularization, and can achieve 1% to 8% improvement over supervised only approaches.

The simplicity of these methods, their robustness, and their high performance give promise that they may have a wide application and impact in use for semi-supervised machine learning.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant # IIS-0326249.

References

S. Abney. Understanding the yarowsky algorithm. *Computational Linguistics*, 30:3, 2004.

Y. Altun, D. McAllester, and M. Belkin. Maximum margin semi-supervised learning for structured variables. In *NIPS*, 2005.

R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6, 2005.

S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: Taking random walks through the view graph. In *WWW*, 2008.

K. Bellare, G. Druck, and A. McCallum. Alternating projections for learning with expectation constraints. In *UAI*, 2009.

Y. Bengio, O. Dellalleau, and N. Le Roux. Label propagation and quadratic criterion. In O. Chapelle, B. Schlkopf, and A. Zien, editors, *Semi-Supervised Learning*. MIT Press, 2006.

A. L. Berger, V. J. Della Pietra, and S. A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), 1996.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincut. In *ICML*, 2001.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

P. F. Brown, V. J. D. Pietra, P. V. DeSouza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, pages 467–479, 1992.

C.J.C Burges and J.C. Platt. Semi-supervised learning with conditional harmonic mixing. In O. Chapelle, B. Schlkopf, and A. Zien, editors, *Semi-Supervised Learning*. MIT Press, 2006.

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

O. Chapelle, B. Scholkopf, and A. Zien. Analysis of Benchmarks. In O. Chapelle, A. Zien, and B. Scholkopf, editors, *Semi-Supervised Learning*. MIT Press, 2006.

M. Chen, I-H. Lee, G. Wu, Y. Wu, and E. Chang. Manifold learning, a promised land or work in progress? In *IEEE/International Conference on Multimedia and Expo*, 2005.

A. Corduneanu and T. Jaakkola. On information regularization. In *UAI*, 2003.

F. Cozman and I. Cohen. Risks of Semi-Supervised Learning. In O. Chapelle, A. Zien, and B. Scholkopf, editors, *Semi-Supervised Learning*. MIT Press, 2006.

Olivier Delalleau, Yoshua Bengio, and Nicolas Le Roux. Large-scale algorithms. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, 2006.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Stat. Soc.*, 39:1–38, 1977.

G. Druck, C. Pal, X. Zhu, and A. McCallum. Semi-supervised classification with hybrid generative discriminative methods. In *KDD*, 2007.

G. Druck, G. S. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 2008.

G. Druck, G. Mann, and A. McCallum. Semi-supervised learning of dependency parsers using generalized expectation criteria. In *ACL/IJCNLP*, 2009a.

G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *EMNLP*, 2009b.

D. Freitag. Trained named entity recognition using distributional clusters. In *EMNLP*, 2004.

K. Ganchev, K. Crammer, F. Pereira, G. Mann, A. McCallum, S. Carroll, Y. Jin, and P. White. Penn/umass/chop biocreativeii systems. In *BioCreativeII*, 2007.

K. Ganchev, J. Gillenwater, and B. Taskar. Dependency grammar induction via bitext projection constraints. In *ACL/IJCNLP*, 2009.

J. Graca, K. Ganchev, and B. Taskar. Expectation maximization and posterior constraints. In *NIPS*, 2008.

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

T. Grenager, D. Klein, and C. Manning. Unsupervised learning of field segmentation models for information extraction. In *ACL*, 2005.

A. Haghighi and D. Klein. Prototype-driven grammar induction. In *COLING-ACL*, 2006a.

A. Haghighi and D. Klein. Prototype-driven learning for sequence models. In *NAACL*, 2006b.

G. Ifrim and G. Weikum. Transductive learning for text classification using explicit knowledge models. In *PKDD*, 2006.

F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *COLING/ACL*, 2006.

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

Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999. URL `citeseer.ist.psu.edu/joachims99transductive.html`.

S. Kakade, Y-W. Teg, and S.Roweis. An alternate objective function for markovian fields. In *ICML*, 2002.

D. Klein and C. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *ACL*, 2004.

M. Kockelkorn, A. Luneburg, and T. Scheffer. Using transduction and multi-view learning to answer emails. In *PKDD*, 2003.

M. Krogel and T. Scheffer. Multi-relational learning, text mining, and semi-supervised learing for functional genomics. *Machine Learning*, 57, 2004.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

J. Lafferty, Y. Liu, and X. Zhu. Kernel conditional random fields:repsentation, clique selection, and semi-supervised learning. In *ICML*, 2004.

W. Li and A. McCallum. A note on semi-supervised learning using markov random fields. Computer science technical note, University of Massachusetts, Amherst, MA, 2004.

W. Li and A. McCallum. Semi-supervised sequence modeling with syntactic topic models. In *AAAI*, 2005.

P. Liang, M. Jordan, and D. Klein. Lerning from measurements in exponential families. In *ICML*, 2009.

S. Macskassy and F. Provost. Classification in networked data. Technical Report CeDER-04-08, New York University, 2006.

R. Malouf. A comparison of algorithms for maximum enotrpy parameter estimation. In *COLING*, 2002.

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

G. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*, 2008.

A. McCallum, K. Bellare, and F. Pereira. A conditional random field for discriminatively-trained finite-state string edit distance. In *UAI*, 2005.

P. Merialdo. Tagging english text with a probabilistic model. *Computational Linguistics*, 1994.

S. Miller, J. Guinness, and A. Zamanian. Name tagging with word clusters and discriminative training. In *ACL*, 2004.

V. Ng and C. Cardie. Weakly supervised natural language learning without redundant views. In *HLT-NAACL*, 2003.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. In *AAAI*, 1998.

K. Nigam, A. McCallum, and T. Mitchell. Semi-supervised Text Classification Using EM. In O. Chapelle, A. Zien, and B. Scholkopf, editors, *Semi-Supervised Learning*. MIT Press, 2006.

Z.-Y. Niu, D.-H. Ji, and C. L. Tam. Word sense disambiguation using label propagation based semi-supervised learning. In *ACL*, 2005.

N. Quadrianto, A. J. Smola, T.S. Caetano, and Q.V. Le. Estimating labels from label proportions. In *ICML*, 2008.

E. Riloff and J. Shepherd. A Corpus-based Bootstrapping Algorithm for Semi-Automated Semantic Lexicon Construction. *Journal for Natural Language Engineering*, 2000. forthcoming.

R. Salakhutdinov, S. Roweis, and Z. Ghahramani. Optimization with em and expectation-conjugate-gradient. In *ICML*, 2003.

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

D. Schuurmans. A new metric-based approach to model selection. In *AAAI*, 1997.

V. Sindhwani and S. S. Keerthi. Large scale semi-supervised linear svms. In *SIGIR*, 2006.

N. Smith and J. Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *ACL*, 2005.

J. Suzuki and H. Isozaki. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *ACL*, 2008.

J. Suzuki, A. Fujino, and H. Isozaki. Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In *EMNLP-CoNLL*, 2007.

Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *NIPS*, volume 14, 2002.

L. Wang, P. Xue, and K. Chan. Incorporating prior knowledge into svm for image retrieval. In *ICPR*, 2004.

S. Wang, R. Rosenfeld, Y. Zhao, and D. Schuurmans. The latent maximum entropy principle. In *IEEE ISIT*, 2002.

J. Weston, C. Leslie, E. Ie, and W. S. Noble. Semi-supervised protein classification using cluster kernels. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, 2006.

D. Yarowsky. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of ACL*, 1995.

T. Zhang and F.J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *ICML*, 2000.

X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, CMU, 2002.

X. Zhu and J. Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *ICML*, 2005.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic mixtures. In *ICML*, 2003.