Sequence Learning from Data with Multiple Labels

Mark Dredze[1], Partha Pratim Talukdar[2], and Koby Crammer[2]

[1] Human Language Technology Center of Excellence
Johns Hopkins University
Baltimore, MD 21211
mdredze@cs.jhu.edu
[2] Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
{partha,crammer}@cis.upenn.edu

Abstract. We present novel algorithms for learning structured predictors from instances with multiple labels in the presence of noise. The proposed algorithms improve performance on two standard NLP tasks when we have a small amount of training data (low quantity) and when the labels are noisy (low quality). In these settings, the methods improve performance over using a single label, in some cases exceeding performance using gold labels. Our methods could be used in a semi-supervised setting, where a limited amount of labeled data could be combined with a rule based automatic labeling of unlabeled data with multiple possible labels.

1 Introduction

Supervised learning requires large amounts of labeled training data but in many real world settings constraints imposed by cost and time for dataset construction lead to a decrease in data quality and quantity. Often times determining a single best label for an instance is difficult, especially in the case of sequence problems. One possible relaxation is to provide multiple possible training labels without choosing a single correct label. For example, multiple annotators can provide contradictory labels or automated systems can provide several good guesses for the correct label. The resulting adjudication of corpora is expensive; selecting the majority label is a good alternative, but introduces label noise.

However, instead of enforcing artificial agreement on the labels – selecting a single label a priori – all likely labels could be used by the learning algorithm. Alternatively, for some tasks, where generating a potential list of likely labels can be done in an unsupervised manner, a supervised learning algorithm could use all likely labels in learning a model.

We develop learning algorithms that are capable of handling instances with multiple possible labels along with an estimate as to the correct label. The resulting trained model tags the test data with a single correct label. Consider the task of named entity recognition where three annotators label a single sequence but two of them mislabel an organization (Figure 1). Instead of training on the majority label (incorrect in this case),

John	*studies*	*at*	*the*	*University*	*of*	*California*	*.*
PER	*O*	*O*	*O*	*ORG*	*ORG*	*ORG (0.33)*	*O*
						LOC (0.67)	

Fig. 1. A named entity training instance with multiple labels and label priors in parenthesis.

we use both labels weighted by their priors. As the model trains on the entire corpus, it can discover that the minority label is actually more probable. It then re-estimates the probabilities of the given labels and trains a new model. Over time, the algorithm shapes the data into a coherent annotation scheme from which it can learn.

In our setting of sequence learning with multiple labels, we are given a set of labels and an indication as to the probability of the given labels being correct (a prior over labels) for each training instance. Our algorithm works in an iterative fashion: first it creates a sequence model trained on all given labels weighted by their priors. Next, it updates the distribution over labels for each training example based on the likelihood assigned by the learned sequence model. This technique discovers correct labels and uses them for training. Previous work constructed an EM style algorithm for learning classification problems with multiple labels [1] and developed a Conditional Random Field model to handle missing data [2]. We extend this work and create a Multi-CRF (Section 3) that models multiple labels per instance.

However, more information i.e. access to multiple labels, need not necessarily improve learning. While our models can handle multiple labels, we ask when does such information benefit learning and when does selecting the most likely label yield superior results? We begin by analyzing these models by changing the quality and quantity of labelings in NLP data. We demonstrate that under the right conditions, our algorithm for modeling multiple labelings improves over a standard CRF.

2 Learning with Multiple Labels

In supervised learning for classification, we are provided with pairs of training examples (x) and labels (y). In the multiple label setting, we are given a set of possible labels instead of a single label for each instance. A learning algorithm for this setting is introduced in [1]. Formally, we are given i.i.d. training data $\mathcal{D} = \{x^{(i)}, S^{(i)}, \pi_y^{(i)}\}_{i=1}^N$, where $x^{(i)}$ is an instance, $S^{(i)}$ is a set of possible labels, and $\pi_y^{(i)}$ is a prior for each label $y \in S^{(i)}$. This allows for a separate prior for each label for every instance. The algorithm models all possible labels for each instance weighted by the probability that each label is correct. Since there is only one correct label for each instance, we want the model to favor a single label. At the same time we do not want the model to stray too far from the provided priors. The following objective function captures this intuition:

$$\ell(\theta) = \sum_{i=1}^{N} \sum_{y \in S^{(i)}} \hat{P}(y|x^{(i)}) \log \frac{\hat{P}(y|x^{(i)})}{\pi_y^{(i)}} - \sum_{i=1}^{N} \sum_{y \in S^{(i)}} \hat{P}(y|x^{(i)}) \log P(y|x^{(i)}, \theta) \quad (1)$$

where $\hat{P}(y|x^{(i)})$ is the estimated label distribution for a given instance $x^{(i)}$ and θ are the parameters of the model. Following our intuitions above, the first term corresponds to the KL divergence between the estimated label distribution and the prior over labels. This ensures that the model's estimates remain close to the given estimates.[1] The second term is the entropy of the data, corresponding to a Maximum Entropy classifier. The entropy term is modified so as to weigh each label by the estimated probability of the label being correct; the classifier is rewarded for using more likely labels. This objective is minimized iteratively using an EM algorithm: the E-step estimates label distributions, $\hat{P}(y|x^{(i)})$, by keeping model parameters fixed, while in the M-step a Maximum Entropy model learns parameters θ that maximize the entropy of the data.The E-step estimates the label distributions as

$$\hat{P}(y|x^{(i)}) = \frac{\pi_y^{(i)} P(y|x^{(i)}, \theta)}{\sum_{y' \in S^{(i)}} \pi_{y'}^{(i)} P(y'|x^{(i)}, \theta)} \tag{2}$$

for all $y \in S^{(i)}$ and 0 otherwise. Therefore, the Maximum Entropy model influences the beliefs about the correct labels subject to the given prior over labels. When $|S^{(i)}| = 1$ $\forall i$, the model reduces to a standard Maximum Entropy model.

This EM algorithm can be viewed as clustering, where each instance has a prior probability of belonging to a cluster. Instances with a single label (prior of 1) are fixed to a cluster and the algorithm clusters the remaining labels into correct and incorrect clusters. A cluster's quality depends on the ability of the CRF model to learn the associated parameters. Additionally, we can view this as self-training, a process whereby a classifier is trained iteratively on its own output. In this case, the E-step relabels the data and ensures that the algorithm's behavior is restricted since instances with a known label cannot be modified.

3 Learning CRFs with Multiple Labels

This probabilistic framework can be extended to sequence models. We are given i.i.d. training data $\mathcal{D} = \{\mathbf{x}^{(i)}, S^{(i)}, \pi_{\mathbf{y}}^{(i)}\}_{i=1}^{N}$, where $\mathbf{x}^{(i)}$ is an instance, $S^{(i)}$ is a set of labels (label sequences), and $\pi_{\mathbf{y}}^{(i)}$ is a set of priors for labels $\mathbf{y}$ for the sequence. Our goal is to learn a model that, given a sequence $\mathbf{x}$, outputs the correct label sequence $\mathbf{y}$. We use a similar objective as before (1) but extend it to sequences:

$$\ell(\theta) = \sum_{i=1}^{N} \sum_{\mathbf{y} \in S^{(i)}} \hat{P}(\mathbf{y}|\mathbf{x}^{(i)}) \log \frac{\hat{P}(\mathbf{y}|\mathbf{x}^{(i)})}{\pi_{\mathbf{y}}^{(i)}} - \sum_{i=1}^{N} \sum_{\mathbf{y} \in S^{(i)}} \hat{P}(\mathbf{y}|\mathbf{x}^{(i)}) \log P(\mathbf{y}|\mathbf{x}^{(i)}, \theta) \tag{3}$$

While classification (Section 2) used Maximum Entropy, we now use a Conditional Random Field (CRF) [3]. A CRF is defined as

$$P(\mathbf{y}|\mathbf{x}^{(i)}) = \frac{1}{Z(\mathbf{x})} \exp(\sum_k \sum_t \theta_k f_k(y_t, y_{t-1}, \mathbf{x})) \tag{4}$$

[1] It can be argued that the model should have the ability to deviate from the given priors without restriction, but our empirical tests found that inclusion of the first term improved model performance.

where $\{f_k(y, y', \mathbf{x})_{k=1}^{K}\}$ are a set of real-valued feature functions. Inserting the CRF model into (3) and applying the $\log$ yields our objective:

$$\ell(\theta) = \sum_{i=1}^{N} \sum_{\mathbf{y}\in S^{(i)}} \hat{P}(\mathbf{y}|\mathbf{x}^{(i)}) \log \frac{\hat{P}(\mathbf{y}|\mathbf{x}^{(i)})}{\pi_{\mathbf{y}}^{(i)}} - \\ \sum_{i=1}^{N} \sum_{\mathbf{y}\in S^{(i)}} \hat{P}(\mathbf{y}|\mathbf{x}^{(i)}) \sum_{t} \sum_{k} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)}) \quad (5)$$

As is typical with CRFs, we add a Gaussian prior to the objective for regularization (not shown). We call the resulting model **Multi-CRF**. The second and third terms in Equation 5 are identical to a standard CRF likelihood except that it contains a weighted sum over all allowed labels of the sequence.[2] Like before, we minimize the objective with an EM algorithm. The E-step is a straight forward modification of the E-step in Section 2, yielding:

$$\hat{P}(\mathbf{y}|\mathbf{x}^{(\mathbf{i})}) = \frac{\pi_{\mathbf{y}}^{(i)} \exp(\sum_k \theta_k f_k(\mathbf{y}, \mathbf{x}^{(i)}))}{\sum_{\mathbf{y}'\in S^{(i)}} \pi_{\mathbf{y}}'^{(i)} \exp(\sum_k \theta_k f_k(\mathbf{y}', \mathbf{x}^{(i)}))}, \ \forall y \in S^{(i)}, \ 0 \text{ otherwise.} \quad (6)$$

Since the normalization constant cancels out of the numerator and denominator the resulting value is easy to compute using the trained CRF. The M-step minimizes the CRF objective using standard optimization techniques (L-BFGS) [4]. The partial derivatives of the model (5) are given by:

$$\frac{\partial \ell}{\partial \theta_k} = \sum_{i=1}^{N} \sum_{\mathbf{y}\in S^{(i)}} \hat{P}(\mathbf{y}|\mathbf{x}^{(i)}) \sum_{t} f_k(y_t, y_{t-1}, \mathbf{x}^{(i)}) \\ - \sum_{i=1}^{N} \sum_{t} \sum_{y,y'} f_k(y, y', \mathbf{x}^{(i)}) P(y, y'|\mathbf{x}^{(i)}) \quad (7)$$

The variables y, y' range over all states of the model. These derivatives are identical to the standard CRF except there is a weighted summation over allowed label sequences.

4 Evaluation

To create an environment for evaluating effects of data properties on learning, we constructed data with multiple labels similar to Jin and Ghahramani [1] who add labels predicted by a naïve Bayes classifier to training instances to create a training set with multiple labels. We produce a similar dataset for sequence learning using a Hidden Markov model (HMM). The alternate labels contain systematic errors, ie. a consistent

[2] We note that $|S^{(i)}|$ (the number of labels) can grow exponentially with the length of the sequence.

labeling, and not random errors. If we randomly permuted labels they would be easy to correct, as noted by the results of Jin and Ghahramani.

Using this approach we created datasets of varying sizes (n) and noise levels (α), where α is defined as the probability that an incorrect label will be given a greater prior than the correct label. n corresponds to the quantity of the data and α to its quality. First, we train an HMM on all available training data and label each training instance with the HMM's prediction, retaining sentences for which the prediction differs from the correct label. Next, we select the first n sentences from the training data to create datasets of varying size. We then assign a prior to each label ($\pi_{\mathbf{y}}^{(i)}$), where the correct label receives a higher prior $(1 - \alpha)$ fraction of the time. We set the prior of the more likely label (max) to be $1 - \alpha$ and the less likely label (min) to α. This ensures that the likelihood that the max label is correct matches the noise level of the data.

We selected two common benchmark sequence labeling tasks in the NLP community for evaluating our algorithms: the CoNLL 2003 English named entity dataset [5] and the CLASSIFIEDS segmentation data [6]. For the CoNLL dataset, we used annotations for people, locations, and organizations and created datasets of $n = 200$, 1k, and 5k using the given 14,042 training sentences with noise levels (α) of 0.1, 0.3, and 0.45. Results were validated on the 3,251 development instances and tested on the 3,454 test instances. [3] For CLASSIFIEDS, where only 103 training instances are provided, we created datasets of $n = 10$, 20, and 50 instances with noise levels (α) of 0.2 and 0.4. Results were validated on the 101 development instances and tested on the 101 test instances. We use standard orthographic features for these types of tasks [7]. For the CoNLL dataset, 5000 training instances resulted in 160k features,while in the CLASSIFIEDS dataset 50 training instances resulted in 25k features.

4.1 Results

We evaluated our CRF learning algorithm against two baselines: a CRF trained on the correct labels (*GOLD*) and a CRF trained on the most likely (maximum) label (MAX). The former indicates performance knowing the correct annotation and the latter is a heuristic used to select from multiple labels, ie. take the best label only. We evaluated several CRF multiple label algorithms on both tasks. First, we ran the Multi-CRF once without the EM algorithm, meaning that it does not reestimate label distributions (*Multi*). Our second test ran the full EM algorithm with the Multi-CRF (Multi_{EM}). We also included a version of the EM algorithm using MAX in the M-step (MAX_{EM}). This can improve over the baseline since it re-estimates the label distributions and relearns. We include this for comparison with Multi_{EM}.

For each experiment, we trained the CRF for 20 iterations, ran 7 EM iterations for MAX_{EM}, and for Multi_{EM}, 50 EM iterations on CoNLL and 30 EM iterations on CLASSIFIEDS. MAX_{EM} converges faster since it does not reestimate label distributions. We observed that the number of iterations mentioned above were enough for convergence. We selected the highest performing model on development data. Results

[3] Setting α as 0.5 would randomize the data; we instead use 0.45 since we assume that the data contains some indication as to the correct annotation scheme.

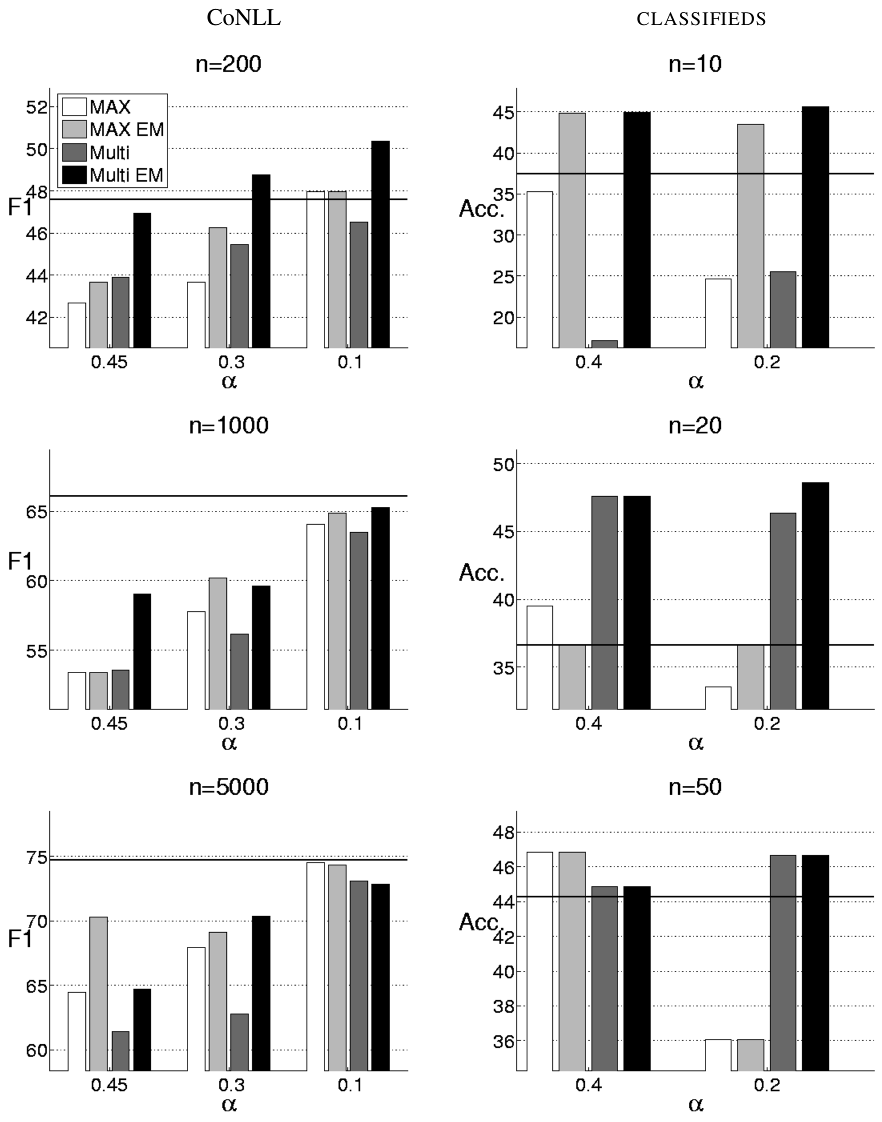

Fig. 2. Performance of the CRF learning methods on CoNLL (left)and CLASSIFIEDS (right) for increasing data set sizes (n) (top to bottom) and noise levels (α) (left to right in each figure). *GOLD* performance is indicated by a horizontal line.

for each dataset size (n) and each noise level (α) are shown in Figure 4.1 for CoNLL and CLASSIFIEDS.

Learning using multiple labels ($Multi_{EM}$) improved over taking the maximum label (MAX) most of the time, even improving over *GOLD* in some cases. While this last

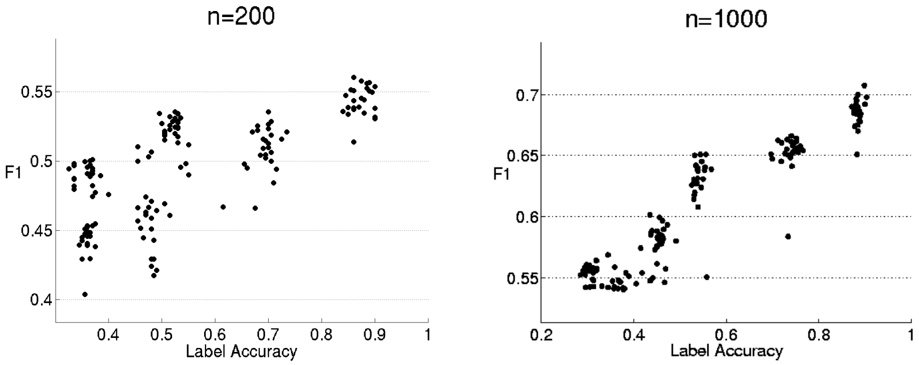

Fig. 3. The x-axis shows the agreement after each iteration between the estimated most likely label and the true gold label in the training data. The y-axis is the resulting F1 score on development data. Data points were taken after each iteration (150 points) for runs of the Multi-CRF on CoNLL $n = 200$ (left) and n =1k (right). Identical label agreements do not necessarily indicate identical clusterings of labels, so they can produce models with different F1 scores.

point seems strange, it could be that allowing the algorithm to influence the data can modify the data so that it is easier to learn, removing difficult examples and improving generalization performance on test data. In this way, our EM algorithm can be considered a clustering algorithm for the labels. In the case of two labels per instance, the algorithm clusters labels into the majority and minority label groups. Figure 3 shows the impact of these clusters on learning. When the labels assigned higher probability by the model are correct (improved cluster accuracy) model performance improves. Maximizing label accuracy tends to maximize F1.

5 When is Learning Successful?

While results show that multi-label learning can improve over learning with a single label, it is helpful to consider when we can expect to see such improvements. Our results indicate that two parameters effect learning: the quantity (n) and quality (α) of the training data. On one end of the spectrum, with small amounts of training data and lots of noise, low quality and quantity, our algorithms give the largest improvements. This makes sense since there is both the greatest potential for improvement (difference between MAX and *GOLD*) and significant information can be gained from the additional labels. This is exactly when one would use our methods. On the other end of the spectrum, with low noise and lots of data, both high quality and quantity, improvements are minimal and our algorithms can even hurt performance. In these cases, there is sufficient training data of high quality that multiple labels are unhelpful.

Between these two extremes, as either quality or quantity improve the baseline (MAX) approaches the performance of gold data. When a model has access to a large number of examples, it can more easily find outliers (noise) by examining many similar instances, allowing good performance with lots of low quality data. The one case

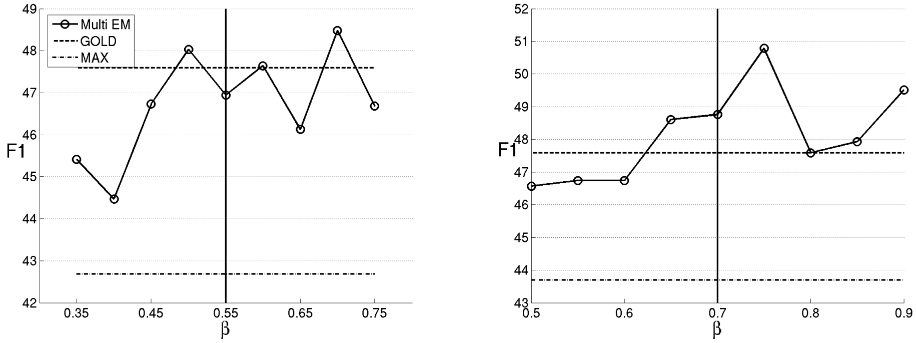

Fig. 4. The performance of a CRF $Multi_{EM}$ on CoNLL with $n = 200$, with α of 0.55 (left) and 0.7 (right). The x-axis contains varying β plotted against the resulting F1 scores. *GOLD* and MAX are shown as horizontal lines and the vertical line indicates the fixed α value.

where MAX outperforms all CRF multiple label methods is where we have the highest quantity ($n = 5000$) and quality ($\alpha = 0.1$) of data.

Noise and reliability of data has been studied in several settings [8]. In a standard theoretical result, Aslam and Decatur give a lower bound on the number of examples needed for learning using a noise level parameter [9]. As noise increases, the number of examples needed increases as well, meaning that an increased training set will offset learning errors from noise.[4] Our empirical observations are consistent with these theoretical results.

Another important point is that the efficacy of learning with multiple labels depends on the accuracy of the noise level belief. Consider the case where the algorithm is provided a majority and minority set of labels for each instance, but is told incorrectly that the majority labels are always correct ($\alpha = 0$). Obviously, the algorithm will ignore the minority labels and not improve. Conversely, if all majority labels are correct and the algorithm is told that $\alpha = .5$, it will incorrectly model the minority labels. Therefore, the algorithm should only consider alternate labels if the majority labels are incorrect, ie. the importance the algorithm assigns to minority labels should be proportional to the likelihood that they are correct.

In the experiments in Section 4.1, we assumed knowledge of the noise level and set the priors of the labels appropriately. We now test the impact of the priors by varying the max label's prior β from the true noise level α. We evaluated a CRF $Multi_{EM}$ varying β from α by 0.20 in increments of .05. Our results (Figure 4) show that different values of β impact performance but in all cases the model still outperforms the MAX baseline. It appears to be safer to be conservative, underestimating the noise level of the data. As more minority labels are ignored, the model reverts to MAX.[5]

[4] This result assumes that the noise in the data is random. While our alternative labels are not random (they are generated by an HMM), we randomly decide if it should have a higher prior than the correct label.

[5] Our observations here about the importance of β apply to the results of Jin and Ghahramani as well. While they do not consider these values, their data uses a β of 2/3 even though α is set to

Our experiments have assumed that all majority labels shared a single prior but our algorithmic formulation allows for a per-instance prior estimate. We tested a CRF $Multi_{EM}$ on a per-instance prior dataset. We selected the CoNLL $n = 200$ dataset and randomly generated an α for each instance between 0.5 and 1. We selected the correct label to be the majority label with a probability of the α for that instance. The prior of each majority label (β) was set to be α plus some random noise (up to 0.1). The resulting dataset had a different prior β for each instance which is a noisy estimate of the true noise level α for each instance. CRF $Multi_{EM}$ improved by approximately 1% over MAX on this data, showing that our algorithms can be competitive even in a per-instance prior setting.

6 Related Work

Most previous work on multi-label classification has focussed on the setting where a single instance can have multiple valid labels [10]. In contrast, the setting considered in this paper involves instances with a single valid label, though during training the instances can have multiple labels assigned to them, at most one of which is correct. This is similar in spirit to that of [1] which deals with learning from multiple labels in the *classification* setting, while the focus of this paper is to learn *structured predictors*.

A closely related area of research studies priors over parameters (instead of labels) [11]. For example, an algorithm for transfer learning by specifying priors over parameters is presented in [12]. Similarly, Raina et al. [11] compute priors over model parameters from multiple users in a transfer setting. These methods rely on the specification of priors over the model parameters, a difficult task for a human annotator. In contrast, we allow for priors over labels, which are easily specified and contain rich information about relevant features.

Recent work on *expectation regularization* (XR) [13] for classification uses a prior over labels in a corpus. This knowledge is combined with unlabeled data for effective semi-supervised learning. While the idea of providing a prior over labels is similar to our setting, there are several important differences. First, XR incorporates priors over single labels (or features), without a clear way of extending the method to interactions between multiple features. In contrast, a prior over a label in our setting can capture multiple feature interactions. Additionally, XR uses a prior over a label *type* at the corpus level, e.g. the user specifies that a particular label, *B-PER*, is likely to occur around 10% of the time in the whole corpus. On the contrary, our method defines a per-instance (and also per-position) prior over labels, which adds further granularity to the description of the data. Finally, since XR adds an additional regularization term to the objective function, we could add a similar term to our Multi-CRF for XR semi-supervised learning with multiple labels in training data. More recently, *Generalized Expectation (GE)* [14, 15] has been proposed using which one can learn from labeled features instead of labeled instances. Such expectation constraints are very useful when one can reliably label features. As in XR, GE expectation constraints are specified across all instances while the multi-label setting considered in this paper is instance specific.

0.7. While their methods appear to do well, they do not include a max label baseline. Without duplicating their experiments, we cannot determine the impact of β on their setting.

7 Conclusion

In this paper we have presented novel learning algorithms for learning structured predictors from multiple labels in the presence of noise. CRF based models improve performance on two standard NLP tasks when we have smaller amount of training data (low quantity) and when the majority labels are noisy (low quality). In these settings, the methods improve performance over using a single max label, in some cases exceeding performance using gold labels. An analysis of these results shows where multi-label learning can be most effective: when data suffers from either low quality or low quantity.

References

1. Jin, R., Ghahramani, Z.: Learning with multiple labels. In: Neural Information Processing Systems (NIPS). (2002)
2. Bellare, K., McCallum, A.: Learning extractors from unlabeled text using relevant databases. In: Workshop on Information Integration on the Web at AAAI. (2007)
3. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning (ICML). (2001)
4. Nocedal, J., Wright, S.: Numerical optimization. Springer (1999)
5. Tjong, E.F., Sang, K., Meulder, F.D.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Conference on Natural Language Learning (CoNLL). (2003) http://www.cnts.ua.ac.be/conll2003/ner/.
6. Grenager, T., Klein, D., Manning, C.: Unsupervised learning of field segmentation models for information extraction. In: Association for Computational Linguistics (ACL). (2005)
7. McDonald, R., Pereira, F.: Identifying gene and protein mentions in text using conditional random fields. BMC Bioinformatics (S6) (2005)
8. Crammer, K., Wortman, J., Kearns, M.: Learning from data of variable quality. In: Neural Information Processing Systems (NIPS). (2005)
9. Aslam, J.A., Decatur, S.E.: On the sample complexity of noise-tolerant learning. Information Processing Letters (1996) 189–195
10. Tsoumakas, G., Katakis, I.: Multi label classification: An overview. International Journal of Data Warehousing and Mining (2007)
11. Raina, R., Ng, A.Y., Koller, D.: Constructing informative priors using transfer learning. In: International Conference on Machine Learning (ICML). (2006)
12. Marx, Z., Rosenstein, M.T., Dietterich, T.G., Kaelbling, L.P.: Two algorithms for transfer learning. In: Workshop on Inductive Transfer: 10 years later at NIPS. (2005)
13. Mann, G.S., McCallum, A.: Simple, robust, scalable semi-supervised learning via expectation regularization. In: International Conference on Machine Learning (ICML). (2007)
14. McCallum, A., Mann, G., Druck, G.: Generalized expectation criteria. Computer science technical note, University of Massachusetts, Amherst, MA (2007)
15. Mann, G., McCallum, A.: Generalized Expectation Criteria for Semi-Supervised Learning of Conditional Random Fields. In: Association for Computational Linguistics (ACL). (2008)