Project Report for 10-707/11-748

Tal Blum, Pradeep Ravikumar, Ashwin Tengli

1 Introduction

Maximum Entropy models are an attractive choice for NLP tasks where informative yet overlapping and dependent features can be defined, in order to enhance extraction. Maximum Entropy models have been shown to outperform HMM models for such tasks [3]. Maximum Entropy Models however are known to have the problem of a *label bias* [2] or more often an *observation bias* [1]. Conditional Random Fields (CRFs) is one model that has been proposed to alleviate this problem [2]. CRFs however are harder to implement and involve iterative optimization over parameters that are normalized over the entire sequence. This results in the training being very slow and also attracts the possibility of overfitting since every parameter is dependent on the rest of the parameters. Several methods have been proposed to learn the CRF parameters more efficiently [4].

We would like to use Maximum Entropy like methods to achieve the same goal of alleviating label and observation bias, but without incurring the optimization cost of CRFs. For the same, we explore an extension to the algorithm of [5].

Motivation:

The optimization function used for selecting the optimal sequence is an important factor in the error rate. Most algorithms use the Viterbi Algorithm - which computes a most probable sequence - for computing the classification sequence. However, for evaluation, in general, the error rate used is a sum of the errors in labeling each tag, instead of the error rate of mis-labelling the entire sequence. This indicates that an optimization function that allows local selection of optimal tags would provide better results for such per-tag accuracy metrics. One example of this would be the optimization function implied by the forward algorithm, which computes the marginal probability of a tag given the observations by summing over the probabilities of the left and right contexts so as to compute the conditional marginal.

However, in graphical models with a conditional structure, there are independencies between states and observations imposed by the graphical model which create problems on using a local optimization function. For example, in a Left to Right MEMM, with observations $\vec{x}$ and labels $\vec{y}$, any state y_i is independent of future observations x_{i+k}. If we locally optimize for the value of

y_i, then the best value that can be output for the label y_i is the mode of its marginal probability conditional on the observations, which not depend on the observations x_{i+1} onwards:

$$Pr(y_i|\vec{x}) = Pr(y_i|x_1, \ldots, x_{i-1})$$

When we perform global optimization by finding the most probable label sequence (Viterbi), the above problem is alleviated since x_{i+1} does affect the output label y_i. This is because while x_{i+1} does not affect the marginal probability of y_i it does affect the probability of the best sequence, and consequently the output y_i also.

But, we want to do a local optimization since we want to optimize the per-tag accuracy. We can do this, and still alleviate the conditional independency problems in the forward approach by introducing bi-directional edges in the conditional model.

2 Stacked Bidirectional MEMM

An MEMM can be trained from left to right (LR) or from right to left (RL). There is no clear preference as to which direction to use for modeling. In practice LR models produce better results. [5] claim that to incorporate bidirectional information one has to maximize $\prod_t P(Y_t|Y_{t-1}, W_t) * P(Y_t|Y_{t+1}, W_t)$. In their implementation they use a adapted version of the viterbi algorithm in which the local term in the viterbi is $P(Y_t|Y_{t+1}, Y_{t-1}, W_t)$ which is learnt using Maximum Entropy estimation techniques.

We propose to extend this framework in two ways - We do a local optimization to select the best label as opposed to a Viterbi-like global optimization. And secondly, we have a hybrid model wherein there is a separate graphical model for each node: The edges are directed left-to-right to the left of the node, and from right-to-left to the right of the node. We can do inference on these many models as follows:

We train a single left-to-right MEMM and a single right-to-left MEMM on the data. We also learn a maximum entropy classifier as in [5] which uses the history and future labels (from both sides) as features.

At each state, we pick the best label value by summing over all left as well as right contexts. We compute the probability of the left context from a left-to-right MEMM and that of the right context by a right-to-left MEMM. The left-to-right and right-to-left MEMMs allow information to pass from obervations on both sides, thus alleviating the problem of marginal independencies (future obervations not affecting a label) in a single directional MEMM.

Thus, our *local* optimization function is

$$SCORE(Y_t) = \sum_{(Y_{t-1}, Y_{t+1})} FORWARD\_SCORE(1, t-1, Y_{t-1}) * P(Y_t|Y_{t-1}, Y_{t+1})$$
$$* BACKWARD\_SCORE(t+1, n, Y_{t+1})$$

where $FORWARD\_SCORE(1, t-1, Y_{t-1})$ is the marginal conditional probability $P(Y_{t-1}|x_1, \ldots, x_{t-1})$ from the left-to-right MEMM. Similarly, $BACKWARD\_SCORE(t+1, n, Y_{t+1})$ is the marginal probability $Pr(Y_{t+1}|x_{t+1}, \ldots, x_n)$ obtained from a right-to-left MEMM.

This model enables explicit belief propagation in both directions in contrast to the to only one direction in the usual MEMM or with only one look-ahead as in Tautanova et al. Each of left-to-right MEMM on the left context and the right-to-left MEMM on the right context enable passing information in a different direction.

3 Experiments

3.1 The Data

For our experiments we chose a sequential classification task of protein secondary structure prediction. Each protein is a sequence of amino acids or residues usually denoted by one letter out of 20 possible residues. Our purpose is to predict the secondary structure of the protein. In our dataset there are 5 possible tags: C-coil, H-alpha helix, B-Beta strand and U, which are probably some special kinds of C-coil. In our experiments we treat those as seperate classes. In addition to the protein sequence there is also experimental data i.e Nucleic Magnetic Resonanse (NMR) measurements done on the protein. Some of these measurements are correlated with the local secondary structure. The data is all real measurements, with all the problems that accompany real data. To name a few - some of the sequence would be missing together with its labels since the people who constructed the database care only for NMR data and did not bother to write residues which have no NMR measurements. The NMR data also has many missing observations. We deal with the missing residues and labels by incorporating a special missing value '*' for a label and for a residue. This makes our classification task to be over 5 classes.

We deal with the missing measurements by incorporating two kinds of features. A numerical feature holding the actual measurement which has value 0 if the measurement is missing and a binary feature which is set to 1 whenever the observation is missing. In addition to the above information, more features were incorporated into the dataset by incorporating residue properties such as the hydrophobia of the residue. Also, spacial filters were applied to the sequence numeric features to catch different known biological secondary structure signatures.

The residues and NMR measurements are only weakly correlated with the secondary structure. The fact that the dataset has so many problems and the fact that we are prediction over 5 classes makes this task very hard. In previous experiments, we got upto an error rate of 34%, however for this project, we chose to run the experiments on only a subset of the data and the possible features due to time limitations.

Method	Noisy (vs True) Label	Sum over histories (vs Max)	Prob. of contexts as Features
Bidirec	-	+	-
BidirecStack	+	+	-
BidirecMax	+	-	-
BidirecProb	+	+	+

Table 1: Method Descriptions

	Error		
	100 Features	300 Features	500 Features
Left-Right MEMM	69.2	64	53.75
Right-Lef MEMM	64.6	64	54.94
CRF	61.5	**57.6**	**49.4**
Bidirec	68	60	53.6
BidirecStack	56.5	61	53.2
BidirecProb	65.4	61	55
BidirecMax	58.2	60.1	53.6

Table 2: Results

3.2 Models

We also incorporated some additions to our basic hybrid-bidirectional model. In the basic model, the central maximum entropy learner learns from the data by using the true tags as history and past features. We incorporated a more stacked approach by training the central maximum entropy learner over the data by using the tags *outputted* by the left-to-right and the right-to-left as the past and future history tags respectively (BidirecStack).

In addition, we also computed the maximum joint probability over the past and future histories instead of summing over them as our score at any node (BidirecMax).

We also tried adding the mode probabilities of the history and future tags from the left-to-right and right-to-left MEMMs as features (BidirecProb).

The various models are summarized in Table 1.

The Table 2 compares the evaluation results of the various methods.

4 Conclusions

Our proposed method clearly outperforms the single directional MEMMs. However, it does not always perform as well as the CRF. It seems that optimizing the parameters over the whole sequence is actually reduces overfitting for the current dataset.

An interesting observation on comparing the results for the BiDirec and BiDirec-Stack, is that we have a bias-variance trade-off. In the BiDirec model, we are training the center Maximum Entropy model based on the true left and right labels; while when testing, we are using the estimated left and right labels from left and right MEMMs. The BiDirect model therefore can not learn the errors of the left-right MEMM and the right-left MEMM. BiDirect-Stack is doing a more optimal thing in that it is learning over the output of the two MEMMs. We denote this output as Noisy Labels as opposed to True Labels. The label features exhibit a bias-variance trade-off. Basically, using the True Labels we have a bias in that we are learning based on something else than what we are tested on. However, the variance of those labels or the quality of them as features is better since they don't contain noise from the errors performed by the MEMMs. When the number of features is small, the bias is much stronger than the variance and therefore the BiDirec-Stack outperforms the BiDirec model. When the number of features increases, the overall prediction accuracy increases and the bias decreases since the MEMMs predict closer to the true labels. This improves the BiDirec model performace. Another factor is that incorporating more features into the logistic function the low variance of the BiDirec makes it harder to overfit compared with the BiDirec-Stack. Therefore, on the second experiment (300 features) the BiDirec outperforms BiDirec-Stack. When we are not overfitting and the labels are less noisy we expect the BiDirec-Stack to again start outperforming BiDirec since its variance is now less. To test this, we ran the models on a larger dataset(not to overfit) with more features(to reduce the noise) and got equality in the error rate between the models. We expect that with a larger database and more features the BiDirec-Stack would continue the trend and be better.

Comparing the BiDirectSum and BidirectMax shows no significant difference in performance. Using the modes of the probabilities of the left and right contexts as features overfitted the model and did not perform well.

References

[1] Dan Klein and Christopher D. Manning. Conditional structure versus conditional estimation in nlp models. In *Stanford Technical Report*, 2002.

[2] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[3] Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598. Morgan Kaufmann, San Francisco, CA, 2000.

[4] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Technical Report CIS TR MS-CIS-02-35*. University of Pennsylvania, 2003.

[5] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*, 2003.