CREATING ENSEMBLE OF DIVERSE MAXIMUM ENTROPY MODELS

Kartik Audhkhasi\textsuperscript{1*}, Abhinav Sethy\textsuperscript{2}, Bhuvana Ramabhadr\textsuperscript{2}, Shrikanth S. Narayanan\textsuperscript{1}
\textsuperscript{1}Signal Analysis and Interpretation Lab (SAIL), University of Southern California, Los Angeles, CA
\textsuperscript{2}IBM T J Watson Research Center, Yorktown Heights, NY
audhkhas@usc.edu, asethy@us.ibm.com, bhuvana@us.ibm.com, shri@sipla.usc.edu

ABSTRACT

Diversity of a classifier ensemble has been shown to benefit overall classification performance. But most conventional methods of training ensembles offer no control on the extent of diversity and are meta-learners. We present a method for creating an ensemble of diverse maximum entropy (\(\partial\text{MaxEnt}\)) models, which are popular in speech and language processing. We modify the objective function for conventional training of a MaxEnt model such that its output posterior distribution is diverse with respect to a reference model. Two diversity scores are explored – KL divergence and posterior cross-correlation. Experiments on the CoNLL-2003 Named Entity Recognition task and the IEMOCAP emotion recognition database show the benefits of a \(\partial\text{MaxEnt}\) ensemble.

Index Terms— Maximum entropy model, classifier diversity

1. INTRODUCTION

Ensembles of multiple experts have out-performed single experts in many pattern classification tasks. Well-known examples include the Netflix Challenge [1], the 2009 KDD Orange Cup [2] and the DARPA GALE program [3]. Dietterich [4] notes three reasons which can explain this. First, an ensemble can potentially have lower generalization error as compared to individual classifiers. Second, the training of most state of the art classifiers (e.g. neural networks) involves solving a non-convex optimization problem. Thus, while the individual classifiers can get stuck in local optima, the ensemble has a better chance to come close to the global optima. Finally, the true decision boundary for the problem at hand may be too complex for a single classifier and an ensemble may better approximate it.

Two popular methods for training classifier ensembles are bagging (bootstrap aggregating) [5] and AdaBoost (adaptive boosting) [6]. Consider a training set \(\mathcal{T}\) containing \(N\) pairs of feature vectors and target variables, \({(x_n, y_n)}_{n=1}^{N}\). Bagging proceeds by sampling \(\mathcal{T}\) with replacement and creating \(M\) bootstrapped data sets \(\mathcal{T}_1,...,\mathcal{T}_M\). The \(m\)th classifier (or regressor) is then trained on \(\mathcal{T}_m\). Given a test feature vector \(x\), results from the \(M\) experts are averaged to yield the estimated target variable. Breiman uses a bias-variance decomposition to prove that in case of regression, the mean squared error of the average regressor is less than or equal to the average mean squared error over the individual regressors. The second method, AdaBoost, works by sequentially training the classifiers in the ensemble. The training data for the \(m\)th classifier, \(\mathcal{T}_m\), is created by weighted sampling from \(\mathcal{T}\), where greater probability mass is assigned to the instances which are misclassified by classifiers \(1,...,m-1\). Freund and Schapire have derived an upper bound on the training error of the ensemble, which indicates that increasing the size of the ensemble in AdaBoost reduces the training error towards zero. AdaBoost can also be viewed as minimizing the exponential loss between the training and predicted label.

As noted in [7], the diversity of classifiers in an ensemble is crucial for its overall performance. Ueda and Nakano [8] consider diversity in an ensemble of regressors and derive a bias-variance-covariance decomposition for the average regressor’s mean squared error. The mean squared error reduces as the pairwise diversity between individual regressors (accounted for by the covariance term) increases. Tumer and Ghosh [9] extend the analysis to classification by treating it as regression over the class posteriors. The additional error of the ensemble over the Bayes optimal error is shown to be dependent on the correlation coefficient between class posteriors.

A typical approach to introduce diversity is to use radically different classifiers and/or feature sets. However, this does not offer explicit control on the extent of diversity achieved. Bagging and boosting also suffer from this issue, and require weak/unstable base classifiers for giving a substantial performance gain. Insipite of the evidence linking diversity and ensemble performance, only a few works deal with explicitly creating diverse classifier ensembles. Negative Correlation Learning [10] involves decorrelating errors from the individual neural networks as part of their training. Another work is DECORA TE [11], a meta-learner where the ensemble is built incrementally with each successive classifier trained on a mix of artificial and natural data. Artificial training instances are labeled contrary to the opinion of the current ensemble.

This paper focusses on training diverse maximum entropy (MaxEnt) models. MaxEnt models are state of the art classifiers in many domains, especially speech and language processing. They possess several desirable properties such as flexibility in adding new features, scalable training, easy parameter estimation and minimal assumptions about the posteriors. The next section discusses our approach for training a diverse MaxEnt (\(\partial\text{MaxEnt}\)) ensemble. We present experiments and analysis on the CoNLL-2003 Named Entity Recognition task and the IEMOCAP emotion recognition database in section 3. Conclusions and scope for future work are presented in section 4.

2. TRAINING A \(\partial\text{MAXENT}\) ENSEMBLE

We first review the standard MaxEnt model to set up the notation. Let \(x \in \mathcal{X}\) and \(y \in \mathcal{Y}\) denote the feature vector and class label respectively. The maximum entropy principle aims to find a probability distribution \(P(y|x)\) with maximum entropy subject to the following first order moment constraints for training data \(\mathcal{T}\):

\[
\sum_{n=1}^{N} E_P\{f_i(x_n, y)\} = \sum_{n=1}^{N} f_i(x_n, y_n) \quad \forall i \in \{1, ..., F\}
\]

where \(f_i\) is the \(i\)th feature - an arbitrary function of \(x\) and \(y\). \(E_P\) denotes the expectation with respect to \(P(y|x_n)\). This problem can
be solved by Lagrange’s method, and the resulting distribution is:

\[ P_\Lambda(y|x) = \frac{\exp(\sum_{i=1}^{F} \lambda_i f_i(x, y))}{\sum_{y \in Y} \exp(\sum_{i=1}^{F} \lambda_i f_i(x, y))} \]  
(2)

where \( \lambda_i \) are the Lagrange multipliers. The log-likelihood function of the MaxEnt model over training data \( \mathcal{T} \) is 1:

\[ \mathcal{L}(\Lambda) = \sum_{n=1}^{N} \sum_{i=1}^{F} \lambda_i f_i(x_n, y_n) - \log Z(x_n) \]  
(3)

where \( Z(x_n) \) is the normalization sum in the denominator of Eq. 2. We note that \( \mathcal{L}(\Lambda) \) is concave in \( \lambda_i \forall i \). Hence a simple gradient ascent, Newton-Raphson or a quasi-Newton method (such as L-BFGS [12]) can be used to find the maximum likelihood parameter estimates. The gradient of \( \mathcal{L}(\Lambda) \) is given as:

\[ \frac{\partial \mathcal{L}(\Lambda)}{\partial \lambda_i} = \sum_{n=1}^{N} \left( f_i(x_n, y_n) - E_P(f_i(x_n, y_n)) \right) \]  
(4)

Our task is to train an ensemble of diverse MaxEnt models. We first study the simpler case of training a MaxEnt model which fits the data well but is diverse with respect to a reference model \( Q_{\Lambda'} \). A natural way to achieve this is to introduce a diversity term in the log-likelihood function as follows:

\[ \mathcal{L}_{tot}(\Lambda) = \mathcal{L}(\Lambda) + \alpha D(P_\Lambda, Q_{\Lambda'}) \]  
(5)

where \( \alpha \geq 0 \) is the diversity weight and \( D(P_\Lambda, Q_{\Lambda'}) \) is the diversity between the two models. As is noted in [13], there are multiple ways to capture diversity between two classifiers. We use two intuitive diversity scores - the Kullback-Leibler (KL) divergence between posterior distributions and negative posterior cross-correlation.

2.1. KL Divergence Diversity

The KL divergence from \( Q_{\Lambda'}(y|x_n) \) to \( P_\Lambda(y|x_n) \) is the following ensemble average:

\[ KL_n(Q_{\Lambda'}||P_\Lambda) = \sum_{y \in \mathcal{Y}} Q_{\Lambda'}(y|x_n) \log \frac{Q_{\Lambda'}(y|x_n)}{P_\Lambda(y|x_n)} \]  
(6)

We did not use \( KL_n(P_\Lambda||Q_{\Lambda'}) \) due to difficulty in interpreting its gradient. Adding this expectation over all instances in the training data, the modified log-likelihood becomes:

\[ \mathcal{L}_{tot}(\Lambda) = \mathcal{L}(\Lambda) + \alpha \sum_{n=1}^{N} KL_n(Q_{\Lambda'}||P_\Lambda) \]  
(7)

While \( \mathcal{L}(\Lambda) \) is concave in \( \Lambda \), \( KL_n(Q_{\Lambda'}||P_\Lambda) \) is convex, attaining a minimum value of 0 at \( \Lambda = \Lambda' \). Thus the overall objective function is neither concave nor convex and one can only hope to obtain locally optimal estimates of \( \Lambda \). Furthermore, KL divergence can potentially approach \( + \infty \), making the objective function unbounded. The gradient of \( \mathcal{L}_{tot}(\Lambda) \) can be written as:

\[ \frac{\partial \mathcal{L}_{tot}(\Lambda)}{\partial \lambda_i} = \sum_{n=1}^{N} \left( f_i(x_n, y_n) - \left[ (1 - \alpha) E_P(f_i(x_n, y)) + \alpha E_Q(f_i(x_n, y)) \right] \right) \]  
(8)

This expression is the same as for a conventional MaxEnt model (Eq. 4), except that a linear combination of the feature expectation under \( P_\Lambda \) and \( Q_{\Lambda'} \) is taken. Increasing \( \alpha \) has the effect of increasing the weight on the expectation from the reference model \( (Q_{\Lambda'}) \). While it seems that KL divergence should succeed in achieving diversity between the models, it can be easily shown that this may not be the case in practice. Let the reference model \( Q_{\Lambda'} \) be trained on data set \( \mathcal{T} \) using features \( \{f_i\} \) by maximizing \( \mathcal{L}(\Lambda') \). Upon convergence of its training, the gradient of \( \mathcal{L}(\Lambda') \) will be zero. Hence:

\[ \sum_{n=1}^{N} E_Q(f_i(x_n, y_n)) = \sum_{n=1}^{N} f_i(x_n, y_n) \quad \forall i \in \{1, ..., F\} \]  
(9)

If \( P_\Lambda \) is trained to be diverse with respect to \( Q_{\Lambda'} \) by maximizing \( \mathcal{L}_{tot}(\Lambda) \) using the same data and feature set, we can substitute the above equation in Eq. 8 and arrive at the following result:

\[ \frac{\partial \mathcal{L}_{tot}(\Lambda)}{\partial \lambda_i} = (1 - \alpha) \sum_{n=1}^{N} \left( f_i(x_n, y_n) - E_P(f_i(x_n, y)) \right) \]  
(10)

Hence the gradients for a MaxEnt and \( \partial \text{MaxEnt} \) are the same up to a scalar multiple. At a local optimum, the parameter estimates will satisfy the same constraint as in the case of a conventional MaxEnt model. This problem with KL divergence can be mitigated to some extent by using distinct training sets or features for \( Q_{\Lambda'} \) and \( P_\Lambda \). However it necessitates the search for another diversity score. The next subsection introduces posterior cross-correlation to this end.

2.2. Posterior Cross-Correlation (PCC) Diversity

Making a simplistic assumption, consider independent random variables \( y_P \sim P_\Lambda(y|x) \) and \( y_Q \sim Q_{\Lambda'}(y|x) \). The conditional probability of them being unequal is:

\[ Pr\{y_P \neq y_Q|x\} = 1 - \sum_{y \in \mathcal{Y}} P_\Lambda(y|x)Q_{\Lambda'}(y|x) \]  
(11)

Thus, negative cross-correlation between the two posterior distributions is a natural diversity score. The modified log-likelihood function can be written as follows:

\[ \mathcal{L}_{tot}(\Lambda) = \mathcal{L}(\Lambda) - \alpha \sum_{n=1}^{N} \sum_{y \in \mathcal{Y}} P_\Lambda(y|x_n)Q_{\Lambda'}(y|x_n) \]  
(12)

This objective function is again neither convex nor concave. However, unlike KL divergence, it has the following finite bounds:

\[ \min_{y \in \mathcal{Y}} Q_{\Lambda'}(y|x_n) \leq \sum_{y \in \mathcal{Y}} P_\Lambda(y|x_n)Q_{\Lambda'}(y|x_n) \leq \max_{y \in \mathcal{Y}} Q_{\Lambda'}(y|x_n) \]

The gradient can be shown to be equal to:

\[ \frac{\partial \mathcal{L}_{tot}(\Lambda)}{\partial \lambda_i} = \sum_{n=1}^{N} f_i(x_n, y_n) - \sum_{n=1}^{N} \left[ (1 - Z_{PQ}(x_n)\alpha) E_P\{f_i(x_n, y_n) \} + Z_{PQ}(x_n)\alpha E_Q\{f_i(x_n, y_n) \} \right] \]  
(13)

where \( PQ_{\Lambda,\Lambda'}(y|x_n) \) is the normalized product distribution:

\[ PQ_{\Lambda,\Lambda'}(y|x_n) = \frac{Q_{\Lambda'}(y|x_n)P_\Lambda(y|x_n)}{Z_{PQ}(x_n)} \]  
(14)

and \( Z_{PQ}(x_n) = \sum_{y \in \mathcal{Y}} Q_{\Lambda'}(y|x_n)P_\Lambda(y|x_n) \) is the normalization constant. The above gradient is similar to the one for KL divergence.
2.3. Sequential Training of a \( \partial \text{MaxEnt Ensemble} \)

Consider the training of an ensemble of \( M \) MaxEnt classifiers \( P_{\Lambda_1}, \ldots, P_{\Lambda_M} \) with corresponding training sets \( T_1, \ldots, T_M \). A simple strategy is to train the ensemble sequentially. Let \( \text{MaxEnt}(T) \) denote a function which trains a conventional MaxEnt model on \( T \) and returns the parameters \( \Lambda \). Let \( \partial \text{MaxEnt}(T, Q_{\Lambda'}, \alpha, \Lambda^0) \) denote a function which trains a \( \partial \text{MaxEnt} \) model on \( T \) with respect to \( Q_{\Lambda'} \) using \( \alpha \) as the diversity weight and \( \Lambda^0 \) as the initial value of the parameters. The sequential training process is as follows:

- Train model 1: \( \Lambda_1 = \text{Maxent}(T_1) \).
- For \( m = 2 \to M \)
  - Initialize: \( \Lambda^0_m = \text{Maxent}(T_m) \).
  - Interpolate models \( 1, \ldots, m-1 \):
    \[ Q(y|x_n) = \frac{1}{m} \sum_{j=1}^{m-1} P_{\Lambda_j}(y|x_n) \]
    \[ \forall y \in Y, n \in \{1, \ldots, |T_m|\} \].
  - Train model \( m \): \( \Lambda_m = \partial \text{MaxEnt}(T_m, Q, \alpha, \Lambda^0_m) \).

Since the objective function is no longer concave, we train a \( \partial \text{MaxEnt} \) model in two passes. The first pass finds the ML estimates of the parameters. The second pass performs \( \partial \text{MaxEnt} \) training using the ML parameters as the starting point. This ensures that L-BFGS converges at a local maxima which is not too far from the ML estimate while ensuring diversity. \( \alpha \) is tuned based on F1 score on a development set. During the test phase, labels from all classifiers in the ensemble are fused by simple plurality. More sophisticated ways of classifier fusion were not experimented with since they are not the focus of this paper.

3. EXPERIMENTS AND RESULTS

The CoNLL-2003 Named Entity Recognition (NER) Task has four types of named entities - persons, locations, organizations and miscellaneous [14]. The English task consists of news wire stories from the Reuters corpus between August 1996 and August 1997. We used binary features from Stanford’s NER system which include word identity, POS tags, word character N-grams etc [15]. Original training, development and evaluation sets were used. Performance was measured in terms of the F1 score for named entity detection [14].

Table 1 shows the F1 scores for ensembles of 5 conventional MaxEnt and \( \partial \text{MaxEnt} \) models using the two diversity scores. Two cases are considered – when the 5 training sets are identical and when they are created by bagging. We can observed that for identical training sets, KL divergence gives almost the same performance as 1 MaxEnt model. The minute difference is due to deliberate smoothing of the posterior distributions to prevent KL divergence from becoming indeterminate. On the other hand, PCC-based \( \partial \text{MaxEnt} \) models give an appreciable increase in performance. In the case of bagged training sets, KL divergence is able to achieve a statistically insignificant performance gain over 5 MaxEnt models. However, PCC-based \( \partial \text{MaxEnt} \) models still perform significantly better. We note that in [16], gradient boosting with 10000 2-level decision trees and Newton-Raphson optimization of the exponential loss was shown to give a similar gain over a MaxEnt model.

Since PCC performs significantly better than KL divergence, we analyse it further. Figure 1 shows the F1 score on the evaluation set with an increasing number of models (1 to 25). The performance of bagging saturates much earlier than the \( \partial \text{MaxEnt} \) ensemble. Thus the relative performance improvement of the \( \partial \text{MaxEnt} \) ensemble increases as the number of models is increased. The performance for \( \alpha_e \) indicates an upper bound on the performance for the \( \partial \text{MaxEnt} \) ensemble. As a final analysis of the \( \partial \text{MaxEnt} \) model with PCC diversity, Figure 2 shows the variation of the development set F1 score and average log-likelihood for an ensemble of 5 models with increasing \( \alpha \). The F1 score increases with \( \alpha \) until around \( \alpha = 1.45 \), after which it starts decreasing again. Furthermore, its behaviour becomes more variable with increasing \( \alpha \) because the optimization problem is become more non-concave. It is interesting to note that the log-likelihood remains practically constant until \( \alpha = 1 \), while the F1 score increases significantly over the same range. The drop in log-likelihood from \( \alpha = 1 \) to 1.45 does not adversely impact the
This paper presented a method to create diverse ensembles of MaxEnt models. Two intuitive diversity scores were explored - KL divergence and negative posterior cross-correlation. Experiments conducted on two classification tasks (the CoNLL-2003 Named Entity Recognition Task and the IEMOCAP emotion classification database) show the advantages of training a $\partial_{\text{MaxEnt}}$ ensemble. It was demonstrated that under reasonable assumptions, KL divergence achieves no gain in performance, while posterior cross-correlation performs significantly better. There are multiple directions for future work. Introduction of a diversity term in the standard MaxEnt model objective function made it non-concave – an undesirable property for optimization. We need to explore ways to train diverse models while retaining concavity. Second, since gradient boosting shows a similar gain over a MaxEnt model (albeit with thousands of models in the ensemble), the link between popular variants of boosting and ensemble diversity needs to be explored. Finally, insight into the choice of diversity scores for a given ensemble and database is required.

5. REFERENCES