A Faster Iterative Scaling Algorithm For Conditional Exponential Model

Rong Jin RONG+@CS.CMU.EDU
Rong Yan YANRONG@CS.CMU.EDU
Jian Zhang JIAN.ZHANG@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Alex G. Hauptmann ALEX@CS.CMU.EDU
Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Abstract

Conditional exponential model has been one of the widely used conditional models in machine learning field and improved iterative scaling (IIS) has been one of the major algorithms for finding the optimal parameters for the conditional exponential model. In this paper, we proposed a faster iterative algorithm named FIS that is able to find the optimal parameters faster than the IIS algorithm. The theoretical analysis shows that the proposed algorithm yields a tighter bound than the traditional IIS algorithm. Empirical studies on the text classification over three different datasets showed that the new iterative scaling algorithm converges substantially faster than both the IIS algorithm and the conjugate gradient algorithm (CG). Furthermore, we examine the quality of the optimal parameters found by each learning algorithm in the case of incomplete training. Experiments have shown that, when only a limited amount of computation is allowed (e.g. no convergence is achieved), the new algorithm FIS is able to obtain lower testing errors than both the IIS method and the CG method.

1. Introduction

Conditional exponential model has been one of the popular conditional models in machine-learning field and has been successfully applied to many different machine-learning problems, such as automatic speech recognition (Rosenfeld, 1996), text classification (Nigam, Lafferty & McCallum, 1999), text segmentation (Beefman, Berger & Lafferty, 1997 & 1999), name identity extraction (Borthwick et. al, 1998) and part-of-speech (POS) tagging (Ratnaparkhi, 1996). One advantage of the conditional exponential model versus the other models is that it is able to combine many correlated input evidences for predicting the class labels without requiring input features to be independent from each other. Furthermore, by assigning high weights to the relevant features and low weights to those irrelevant ones, the conditional exponential model can be quite resilient to the introduction of irrelevant features. Another interesting aspect of the conditional exponential model is that it is strongly associated with Maximum Entropy (ME) model (Jelinek, 1997; Berger, Pietra & Pietra, 1996). More precisely, it has been shown that the conditional exponential model is actually a dual problem of ME model and therefore has the unique global maximum.

To find the optimal conditional exponential model for given training data, two groups of approaches have been used in the past research. One is named iterative scaling approach (Brown, 1959), including the Generalized Iterative Scaling (GIS) (Darroch & Ratcli, 1972) and the Improved Iterative Scaling (IIS) (Berger, 1997). The underlying idea for iterative scaling approaches is similar to the idea of Expectation-Maximization (EM) approach: by approximating the log-likelihood function of the conditional exponential model as some kind of 'simple' auxiliary function, the iterative scaling methods are able to decouple the correlation between the parameters and the search for the maximum point can be operated along many directions simultaneously. By carrying out this procedure iteratively, the approximated optimal point found over the 'simplified' function is guaranteed to converge to the true optimal point due to the convexity of the objective function. The distinction between GIS and IIS is that the GIS method requires the sum of input features to be a constant over all the examples while the IIS method doesn't. This constraints can limits the application of GIS, particularly when the sum of features is not bounded. Furthermore, in the previous studies, people have found that the IIS method is able to

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

find the optimal parameters for the conditional exponential model significantly faster than the GIS method, particularly for the applications of natural language processing. Therefore, we will only consider the IIS method as the comparison peer for the proposed method.

The second group of approaches are mainly the generic approaches for nonlinear optimization, including the conjugate gradient approach (CG) (Shewchuk, 1994) and the quasi-Newton method (Liu & Norcedal, 1989). Previous studies have shown that both the conjugate gradient and the quasi-Newton method are able to find the optimal parameters for the conditional exponential model much faster than the iterative scaling methods (Minka, 2001;Malbouf, 2002). One advantage of the conjugate gradient approach versus the quasi-Newton approach is that the quasi-Newton method requires the explicit computation of the approximate Hessian matrix while the conjugate gradient approach does not. Since the number of elements within a Hessian matrix is equal to the square of the number of parameters for the problem, the storage of a Hessian matrix can be extremely expensive if the problem involves hundreds of thousands of parameters. Therefore, in the paper, we will only include the conjugate gradient approach for the comparison.

In this paper, we propose a new iterative scaling method, which shares the similar idea with the previous iterative scaling methods, namely for each iteration, approximating the original log-likelihood function with a lower bound auxiliary function and find the optimal point over the auxiliary function as the approximation to the global optimal solution for the original log-likelihood function. The only difference between this work and previous work on iterative scaling methods is that, this work provides a better approximate auxiliary function that is able to bounds the original log-likelihood function tighter. Therefore, we would expect this new iterative scaling method is able to converge to the global optimium faster than the previous iterative scaling methods. Empirical studies on the text classification over three different collections have shown that the new iterative scaling method is able to achieve significantly faster convergence rate than IIS method. Furthermore, we compared the proposed iterative scaling method to CG method over the same testbed and found that the new algorithm also runs faster than the CG method over all three collections.

The rest of the paper is arranged as follows: The formal description and analysis of the new iterative scaling method will be presented in Section 2. The empirical studies on the effectiveness of the new iterative algorithm are presented in Section 3. Within it, we will examine the convergence rate of the new algorithm with respect to IIS method and CG method, respectively. Conclusion and future works will be presented in Section 4.

2. A Faster Iterative Scaling Algorithm: FIS

As already mentioned in the introduction section, the basic idea of the IIS algorithm is to approximate the log-likelihood function with a lower bound auxiliary function and compute the optimal point over the auxiliary function as the approximation of the global optimal solution for the true log-likelihood function. Therefore, a lower bound auxiliary function that is able to bound the log-likelihood function tighter than the IIS method will lead to a faster convergence rate. In order to propose a better lower bound auxiliary function, we first examine how the IIS algorithm bounds the log-likelihood function in Section 2.1 and then a better auxiliary function is proposed in Section 2.2.

For the sake of simplicity, throughout the rest of this paper, we assume that all the features are nonnegative.

2.1 Overview of Improved Iterative Scaling (IIS) Algorithm

The key component for a conditional model is to compute the $p(y|x)$, namely the likelihood for an instance to have a class label y given the input x. For a conditional exponential model, $p(y|x)$ is usually written as:

$$p(y \mid x) = \frac{1}{Z(x)} e^{\sum_i \lambda_i f_i(x,y)} \tag{1}$$

where $f_i(x,y)$ stands for the i[th] feature extracted from the input x and the output y, and λ_i stands for the corresponding weight. Symbol $Z(x)$ is the normalization constant, which enforces the sum of $p(y|x)$ over different class labels y to be one, i.e.

$$Z(x) = \sum_y e^{\sum_i \lambda_i f_i(x,y)}$$

For the purpose of simplicity, let's assume that every class uses the same set of features $\{f_i(x)\}$. Under that assumption, the general form (1) can be rewritten as

$$p(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_i \lambda_{y,i} f_i(x)\right) \tag{1'}$$

As seen from Equation (1'), the weight $\lambda_{y,i}$ has two indices, with y for the class index and i for the feature index.

The goal of the training procedure is to find the set of weights $\{\lambda_{y,i}\}$ that maximizes the log-likelihood of the training data. Given the empirical data distribution $\tilde{p}(x,y)$ obtained from the training examples, the log-likelihood will be written as:

$$\begin{aligned} L &= \sum_{x,y} \tilde{p}(x,y) \log p(y \mid x) \\ &= \sum_{x,y} \tilde{p}(x,y) \sum_i \lambda_{y,i} f_i(x) - \sum_x \tilde{p}(x) \log\left(\sum_y e^{\sum_i \lambda_{y,i} f_i(x)}\right) \end{aligned} \tag{2}$$

where $\tilde{p}(x)$ stands for the empirical data distribution for input x. Since directly optimizing Equation (2) can be difficult, people take the iterative approach, namely dividing the procedure of maximization into many steps and each iteration will only increase the log-likelihood slightly from the previous iteration. Let $\delta_{y,i}$ stands for change in the weight $\lambda_{y,i}$ between two consecutive iterations. Then, the difference in the log-likelihood L for two consecutive iterations will be expressed as the function of $\delta_{y,i}$, which is

$$\Delta L = \sum_{x,y} \tilde{p}(x,y) \sum_i \delta_{y,i} f_i(x) - \sum_x \tilde{p}(x) \log \left(\sum_y p(y \mid x) e^{\sum_i \delta_{y,i} f_i(x)} \right) \quad (3)$$

As seen from Equation (3), the complexity in maximizing ΔL comes from the second term where the set of parameters $\{\delta_{y,i}\}$ are coupled with each other through the exponential function and the logarithm function. The IIS method uses the inequality $-\log \alpha \geq 1-\alpha$ to decouple the interaction between parameters $\{\delta_{y,i}\}$ due to the logarithm function and Jensen's inequality, namely $\sum_x \exp(p(x)q(x)) \leq \sum_x p(x) \exp(q(x))$ for any p.d.f. $p(x)$, to decouple the correlation caused by the exponential function. With these two inequalities, the resulted lower bound auxiliary function for ΔL is written as:

$$\Delta L \geq Q_{IIS}(\{\delta_{y,i}\}) = 1 + \sum_{y,i} \left\{ \sum_x \tilde{p}(x,y) \delta_{y,i} f_i(x) - \sum_x \tilde{p}(x) p(y \mid x) \frac{f_i(x)}{f^{\#}(x)} e^{\delta_{y,i} f^{\#}(x)} \right\} \quad (4)$$

where symbol $f^{\#}$ stands for the sum of all the features, i.e. $\sum_i f_i(x)$. With the inequality (4), instead of optimizing the true log-likelihood function ΔL, we can maximize the auxiliary function Q_{IIS}. Since the auxiliary function Q_{IIS} has all the interaction between variables $\{\delta_{y,i}\}$ removed, we can simply optimize it with respect to each variable $\delta_{y,i}$ independently from other variables. Furthermore, since Q_{IIS} low bounds the difference in log-likelihood function ΔL, by maximizing Q_{IIS}, we can make sure ΔL to be at least non-negative, which means that the log-likelihood function will never decrease in the iterative procedure. One of the usual procedure used for optimizing Q_{IIS} is the univariate Newton method.

2.2 A New Low Bound Auxiliary Function

2.2.1 Basic Idea

The lower bound auxiliary function Q_{IIS} for the IIS method in Equation (4) can be rewritten as a sum of a set of functions $\{g_{y,i}\}$ and each function $g_{y,i}$ only depends on a single variable $\delta_{y,I}$, i.e.,

$$g_{y,i}(\delta_{y,i}) = \sum_x \tilde{p}(x,y) \delta_{y,i} f_i(x) - \sum_x \tilde{p}(x) p(y \mid x) \frac{f_i(x)}{f^{\#}(x)} e^{\delta_{y,i} f^{\#}(x)}$$

Therefore, the correlation between variables $\{\delta_{y,i}\}$ has been completely decoupled in the IIS method. However, the price paid for the full decoupling is that, the auxiliary function may not be able to bound the original log-likelihood function tightly enough. In particular, inequalities $-\log \alpha \geq 1-\alpha$ and $\sum_x \exp(p(x)q(x)) \leq \sum_x p(x) \exp(q(x))$ have been used in the IIS method. As a result, many iterations are required in order to reach the true optimal solution. Clearly, there is a tradeoff between the complexity of auxiliary function and number of iteration. On one hand, by bounding the log-likelihood function with a simple auxiliary function, we are able to obtain the optimal solution over the auxiliary function quickly however we may have to run through the iterative procedure many times. On the other hand, a complicated auxiliary function may be able to bound the log-likelihood function more tightly however computing the optimal solution of the complicated bounding function may be expensive.

The basic idea of improving the IIS algorithm is to introduce an auxiliary function, which only decouples part of the interaction between parameters. Unlike the IIS method, where Q_{IIS} consists of functions only with a single variable, the new auxiliary function will be the sum of functions $\{g_i\}$ related to multiple variables. By keeping some of the interaction between variable alive in the approximation, we are able to achieve an auxiliary function that bounds the original log-likelihood function more tightly than the IIS algorithm. Meanwhile, only a small number of variables are related to each g_i in the auxiliary function. Therefore the optimization of each function g_i can still be solved efficiently by using traditional numerical methods such as the method of multivariate Newton.

2.2.2 Holder Inequality

The most critical component in the derivation of the proposed auxiliary function is so called 'Holder Sum Inequality' (Abramowitz & Stegun, 1972). In this subsection, we will give a brief introduction of this inequality and its extension.

The original version of Holder Inequality can be stated as follows:

For a set of non-negative variables $\{a_k\}_{k=1}^{n}$ and $\{b_k\}_{k=1}^{n}$, the following inequality will always hold

$$\sum_{k=1}^{n} a_k b_k \leq \left(\sum_{k=1}^{n} {a_k}^p \right)^{1/p} \left(\sum_{k=1}^{n} {b_k}^q \right)^{1/q} \quad (5)$$

for any $p>1$, $q>1$ and $p^{-1}+q^{-1}=1$. Clearly, Cauchy Inequality is a special case of this inequality when both p and q are set to be 2.

Furthermore, the Holder Inequality can be extended to a more general form. Considering the function form $\sum_i \prod_k \alpha_{i,k}$, where all variable $\alpha_{i,k}$ are nonnegative. It is not difficult to show that the following inequality will always hold, i.e.

$$\sum_i \prod_k \alpha_{i,k} \le \prod_k \left\{ \sum_i \alpha_{i,k}^{q_k} \right\}^{1/q_k} \tag{6}$$

for any set of $\{q_k\}$, as long as all the q_k are positive and satisfies the constraint $\sum_k q_k^{-1}=1$. A proof of the extension of Holder Inequality is provided in the Appendix.

To understand why inequality in (6) is useful in building up auxiliary function, we can simply take the logarithm of the both sides of the inequality, i.e.

$$\begin{aligned} \log\left(\sum_i \prod_k \alpha_{i,k} \right) &\le \log\left(\prod_k \left\{ \sum_i \alpha_{i,k}^{q_k} \right\}^{1/q_k} \right) \\ &= \sum_k \frac{1}{q_k} \log\left(\sum_i \alpha_{i,k}^{q_k} \right) \end{aligned} \tag{6'}$$

On the LHS of the above inequality, we have all the variables $\alpha_{i,k}$ couple due to the existence of product and the logarithm function. However, on the RHS of this inequality, we only have $\alpha_{i,k}$ with same index k interacted with each other and all the other couplings between variables $\{\alpha_{i,k}\}$ are removed from the object function. Therefore, by applying inequality (6), we are able to delete part of the interaction between variables so that the original optimization problem is simplified.

2.2.3 A FASTER ITERATIVE SCALING ALGORITHM: FIS

Now consider how to apply the extension of Holder Inequality to find a better lower bound auxiliary function for the log-likelihood function in (3). Notice that, the most complicated term within (3) is term $\log\left(\sum_y p(y|x)\exp(\sum_i \delta_{y,i} f_i(x))\right)$. Similar to the IIS algorithm, let $f^{\#}(x)$ stands for the sum of all the features. If we denote $\exp(\delta_{y,i} f_i(x) + \log(p(y|x)) f_i(x)/f^{\#}(x))$ as $\alpha_{y,i}$, the log-likelihood function can be rewritten as:

$$\begin{aligned} &\log\left(\sum_y p(y|x)\exp(\sum_i \delta_{y,i} f_i(x))\right) = \\ &\log\left(\sum_y \prod_i p(y|x)^{\frac{f_i(x)}{f^{\#}(x)}} \exp(\delta_{y,i} f_i(x)) \right) = \\ &\log\left(\sum_y \prod_i \alpha_{y,i} \right) \end{aligned}$$

Then, according to the extension of Holder Inequality, we have an upper bound for the log-likelihood function as:

$$\begin{aligned} &\log\left(\sum_y p(y|x)\exp(\sum_i \delta_{y,i} f_i(x))\right) = \log\left(\sum_y \prod_i \alpha_{y,i} \right) \\ &\le \sum_i \frac{1}{q_i} \log\left(\sum_y \alpha_{y,i}^{q_i} \right) \\ &= \sum_i \frac{1}{q_i} \log\left(\sum_y p(y|x)^{\frac{q_i f_i(x)}{f^{\#}(x)}} e^{q_i \delta_{y,i} f_i(x)} \right) \end{aligned} \tag{7}$$

By setting $q_i = f^{\#}(x)/f_i(x)$, we have Equation (7) simplified as:

$$\begin{aligned} &\log\left(\sum_y p(y|x)\exp(\sum_i \delta_{y,i} f_i(x))\right) \\ &\le \sum_i \frac{1}{q_i} \log\left(\sum_y p(y|x)^{\frac{q_i f_i(x)}{f^{\#}(x)}} e^{q_i \delta_{y,i} f_i(x)} \right) \\ &= \sum_i \frac{f_i(x)}{f^{\#}(x)} \log\left(\sum_y p(y|x) e^{\delta_{y,i} f^{\#}(x)} \right) \end{aligned} \tag{7'}$$

By substituting Equation (7') for the term $\log\left(\sum_y p(y|x)\exp(\sum_i \delta_{y,i} f_i(x))\right)$ in Equation (3), we will have the following inequality

$$\begin{aligned} \Delta L &\ge Q_{FIS}(\{\delta_{y,i}\}) \\ &= \sum_i \left\{ \begin{array}{l} \sum_{x,y} \tilde{p}(x,y)\delta_{y,i} f_i(x) - \\ \sum_x \tilde{p}(x) \frac{f_i(x)}{f^{\#}(x)} \log\left(\sum_y p(y|x) e^{\delta_{y,i} f^{\#}(x)} \right) \end{array} \right\} \\ &= \sum_i g_i \end{aligned} \tag{8}$$

According to Equation (8), the auxiliary function Q_{FIS} is a sum of a set of functions g_i and each g_i only involved in variables $\delta_{y,i}$ with same feature index i. In general, the number of features is substantially larger than the number of classes. Therefore, the lower bound auxiliary function in (8) is able to remove most of the correlation between variables and only the interaction between variables $\delta_{y,i}$ with the same feature index i are kept. Furthermore, it is not difficult to see that each function g_i within Equation (8) is a convex function by simply checking if the Hessian matrix of each function g_i is semi-positive definite. Therefore, simple methods such as a Newton method can be employed for finding the optimal solution of function g_i because each g_i function only contains a small number of parameters (equal to the number of classes).

Comparing Q_{FIS} in (8) to Q_{IIS} in (4), we can show that Q_{FIS} is an upper bound of Q_{IIS} by using inequality $-\log\alpha \ge 1-\alpha$. The sketch of proof is shown in Equation (9). Therefore, the new iterative scaling algorithm forms a tighter lower bound for the original log-likelihood function. As a result, we would expect the new algorithm is able to converge to the global optimal solution in a significantly smaller number of iteration than the IIS method. However, as mentioned before, the computation complexity of each iteration

and the number of iteration form a tradeoff pair. The new algorithm may use a smaller number of iterations but each iteration could consume more computation cycles. In order to account for the total amount of computation complexity, in the experiment, we simply use the total amount of time consumed CPU by both algorithms, which can be obtained by the matlab command 'cputime'. The algorithm that is able to find the set of good parameters within a smaller amount of CPU time is deemed as a faster algorithm.

$$
\begin{aligned}
Q_{FIS}(\{\delta_{y,i}\}) &= \sum_i \left\{ \begin{array}{l} \sum_{x,y} \tilde{p}(x,y)\delta_{y,i} f_i(x) - \\ \sum_x \tilde{p}(x) \frac{f_i(x)}{f^{\#}(x)} \log\left(\sum_y p(y \mid x) e^{\delta_{y,i} f^{\#}(x)} \right) \end{array} \right\} \\
&\geq \sum_i \left\{ \begin{array}{l} \sum_{x,y} \tilde{p}(x,y)\delta_{y,i} f_i(x) + \\ \sum_x \tilde{p}(x) \frac{f_i(x)}{f^{\#}(x)} \left(1 - \sum_y p(y \mid x) e^{\delta_{y,i} f^{\#}(x)} \right) \end{array} \right\} \\
&= 1 + \sum_i \sum_{x,y} \left\{ \tilde{p}(x,y)\delta_{y,i} f_i(x) - \tilde{p}(x) p(y \mid x) e^{\delta_{y,i} f^{\#}(x)} \frac{f_i(x)}{f^{\#}(x)} \right\} \\
&= Q_{IIS}
\end{aligned}
\tag{9}
$$

Finally, it may be attractive to think that, the inequality (8) can be obtained by simply applying Jensen inequality to both the exponential function and logarithm function. However, that is not true because the Jensen inequality for logarithm function only leads to a lower bound for logarithm function, namely

$$\sum_x \log(p(x)q(x)) \geq \sum_x p(x)\log(q(x))$$

for any p.d.f p(x). Instead, in the above derivation, we need an upper bound function for logarithm as illustrated in Equation (7'). Therefore, using Jensen inequality for logarithm won't lead to the results in (8).

3. Experiments

3.1 Experiment Design

The goal of this experiment is to examine the efficiency of the proposed algorithm on the text classification task. The efficiency issue involves in two aspects:

1) *Whether the proposed algorithm is able to increase the log-likelihood function more efficiently than other learning algorithms?* As pointed out before, though the new algorithm yields a better auxiliary function that bounds the original log-likelihood function more tightly than the IIS method, it may still not be as efficient as the IIS method because in the new algorithm each iteration consumes more computation cycles. Therefore, we need to examine whether the introduction of a tightly bound but complicated auxiliary function is worthwhile. Furthermore, since the conjugate gradient method has been shown to be more efficient than the iterative scaling methods in some of previous studies, we will also compare the proposed iterative scaling method to the conjugate gradient method.

2) *Whether the solution found by the proposed method results in lower classification error when the model is not full trained?* In many cases, due to the limitation of time, we may have to stop the learning algorithm when it is still far from the global optimal solution. Under that circumstance, we need to know the quality of the parameters found by the learning algorithm. A learning algorithm is preferred when it is able to find 'decent' parameters that result in low testing errors even it is far from the convergence point. Notice that it is not always true that parameters that result in a larger value of log-likelihood function of training data will definitely lead to a lower testing error, particularly in case that the model is not fully trained.

For the first efficiency issue, we compute the value of the likelihood function versus the accumulative CPU time for every iteration. An algorithm that is able to achieve a large value of log-likelihood function within a small amount of CPU time is deemed to be a good algorithm. To determine the quality of learned parameters in the middle of learning process, for every 20 iterations, we compute the classification errors on a separate testing dataset using the learned parameters. An algorithm that achieves lower testing error within smaller amount of CPU time is believed to be a better algorithm. Three collections of text classification are used in this experiment and for each collection. For each collection, we split it into a training set and a testing set by 70% vs. 30% shares. The details are described in Table 1.

Data Set	# Vocabulary	#Class	#Training	#Testing
WebKB	19676	6	2398	1355
Industry Sector	28915	38	2215	1021
NewsGroup	47411	11	7149	3298

Table 1: Description of datasets

For comparison methods, we will mainly compare the proposed algorithm to the previous iterative scaling algorithm, namely the IIS method, because both of them use very similar techniques except for the auxiliary function. Meanwhile, previous studies on the conditional exponential model have indicated that the CG method appears to be more efficient than IIS algorithm for learning an exponential model (Minka, 2001; Malbouf, 2002). Therefore, we will also compare the proposed algorithm with respect to the CG method.

In terms of implementation, we try to make each algorithm as efficiently as possible. For the IIS

algorithm, the key computation complexity is on the optimization of the auxiliary function Q_{IIS} in (4). Usually, a uni-variate Newton method is used for finding the optimal solution over Q_{IIS}. Since Newton method is an iterative method, it usually requires at least several iterations to find the optimal solution over the auxiliary function. However, it may not be worthwhile to find the optimal point over the auxiliary function since it is just an approximation of the original log-likelihood function and our goal is to find the optimal solution over the likelihood function not the auxiliary function. In fact, as long as the solution $\delta_{y,i}$ in (4) is able to increase the log-likelihood, the whole iterative scaling method is guaranteed to find the global optimal solution. Therefore, in practice, instead of running the Newton method through many iterations, we simply run it once over the auxiliary function. Furthermore, a linear search is applied in order to guarantee that the new point found in each iteration is always better than the previous one. Our empirical studies have found that this implementation is able to find the global optimal solution substantially faster than the implementation of running the Newton method till it converges. The same strategy applies to the implementation of the proposed FIS algorithm. For the CG method, the choice of search direction has great impact on the convergence speed. In our implementation, we choose Hestenes-Stiefel (Moller, 1993) method since it has been found very efficiently in practice.

3.2 Conditional Exponential Model for Text Classification

The conditional exponential model has been found to be an effective method for text classification in the previous study (Nigram et al., 1999). The main idea is to treat each unique word as a separate feature and try to find the appropriate weights of words for different classes using the conditional exponential model. In addition to the standard practice for conditional exponential model, two main issues need to be considered for the case of text classification:

1) *Feature selection*. As indicated in Table 1, the vocabulary size of each collection is considerably large, around the order of 10,000. Apparently, most of words will not be informative to indicate the category of documents. Thus, it is important to remove those uninformative words and only leave the informative words as the representation features. We use Information Gain (Nigram et al., 1999) as the feature selection criterion, and the top 300 features with the highest information gain are selected. For each feature, the corresponding unigram probability, namely the term frequency of the corresponding word divided by the document length, is used as its value. In addition, we also conducted the same experiments but with top 500 and 1000 selected features. The results are extremely similar to the experiment with only 300 selected features. Due to the limited space, we will only show the results for 300 features.

2) *Regularization*. The conditional exponential model sometimes can give overly large weights to words, particularly those rare words. Consider the case that a word only appears in one document within the whole training corpus. According to the conditional exponential model, this word can have an infinitive large weight. However, this is definitely undesirable since the word may be accidentally used for that document and may not be informative at all. A general practice to avoid this kind of disaster is to introduce some kind of regularization factor. For text classification, people have tried the Gaussian prior as the regularization factor and found it is quite effective (Nigram et al., 1999), which prevents weights from growing too large. Furthermore, people have found that by introducing the regularization factor into the conditional exponential model, we are able to even improve the classification accuracy. We use the similar regularization approach for all the learning algorithms to be compared.

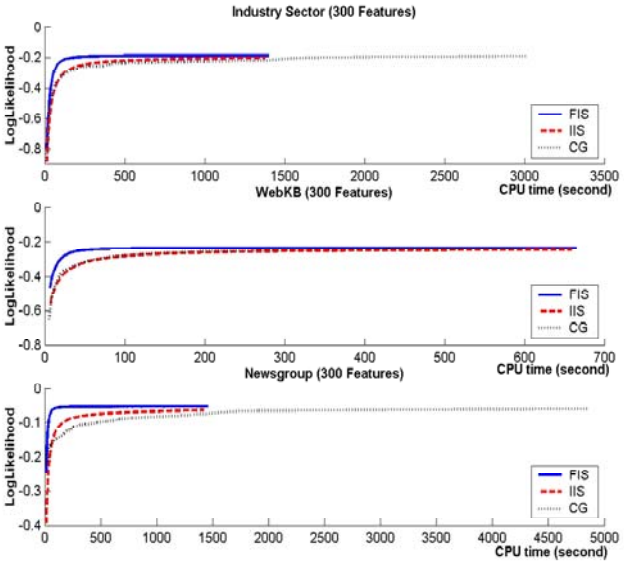

Figure 1: Convegence behaviors of the proposed iterative scaling algorithm FIS and other algorithms namely the IIS and CG method.

3.3 Results And Discussions

3.3.1 Comparison of Convergence Speed

Figure 1 shows the results for IIS, CG and the proposed method FIS over three datasets for text classification. Among all the diagrams, the horizontal axis is the number of seconds used by CPU and the vertical axis represents the log-likelihood. The parameters are

initialized to be zeros for all three algorithms. Due to the large negative values of log-likelihood for the first several iterations (on the order –log(num_of_class)), we only show the curve of log-likelihood since the 20 iterations. The same strategy applies to the Figure 2, when the curve of testing errors is displayed. Clearly, the curve of the FIS algorithm is able to reach the maximum of likelihood much more quickly than the other two algorithms. In addition, the experiments with 500 and 1000 selected features are conducted and the similar behavior is observed, namely the FIS algorithm reaches the maximum of likelihood much faster than both the IIS method and CG. These observations indicate that in the text classification task, the proposed algorithm 'FIS' is a more efficient algorithm in learning the conditional exponential model than both the IIS algorithm and the CG algorithm.

The other interesting observation from Figure 1 is that, the IIS algorithm performs at least as well as the CG algorithm over all the three datasets, which is quite different from what other researchers have claimed (Minka, 2001; Malouf, 2002). We think that it can be attributed either to the special characteristics of the text classification task or to the particular implementation of IIS algorithm used in this paper such as how to find the optimal point over the auxiliary function.

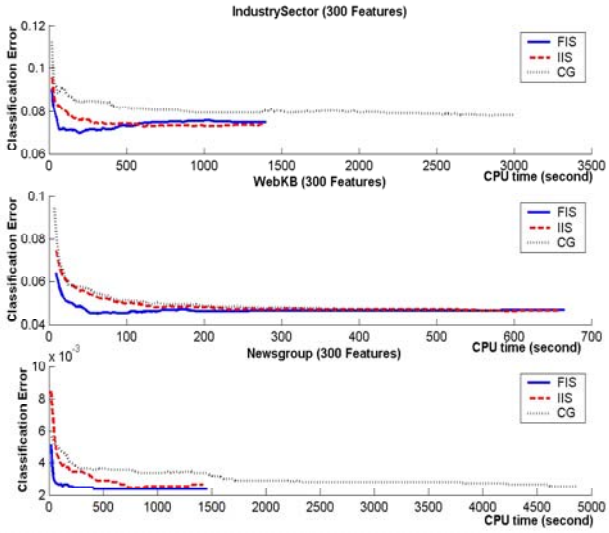

Figure 2: The behavior of testing errors with respect to the running time for the proposed algorithm FIS and other algorithms namely IIS and CG methods.

3.3.2 Quality of Learned Model In Case of Incompletely Training

In addition to the convergence rate, we are also concerned with the quality of parameters learned from the algorithm particularly when the model is not fully trained. Figure 2 plots the behavior of the testing errors with respect to the amount of CPU time devoted to computing. Similar to the previous experiment, parameters are set to be zeros for all three algorithms and the plotted curves start from 20 iterations due to the large testing errors at the beginning of the learning.

According Figure 2, the FIS algorithm is able to reach the lower classification error much faster than the other two algorithms. Meanwhile, for the collection 'Industry Sector', we can see the overfitting problem for the FIS algorithm. By varying the regularization constant, we are able to avoid the overfitting problem but obtain the same classification error at the end. This fact of overfitting in the FIS algorithm indicates the importance of regularization in learning the conditional exponential model. The same experiment with 500 and 1000 selected features are conducted and the similar behaviors are observed, namely the FIS algorithm is able to achieve lower testing errors faster than the IIS algorithm and the CG algorithm. Thus, we conclude that the proposed algorithm FIS is able to not only optimize the log-likelihood function faster than the other two algorithms but also find 'decent' parameters faster. The other interesting observation is that, for collection 'Industry Sector', according to Figure 1, it seems that both the IIS and the CG algorithms have very similar behavior in the convergence of log-likelihood. However, according to Figure 2, for most of time, the IIS algorithm appears to achieve lower testing errors than the CG algorithm in collection 'Industry Sector'. This fact again indicates that a larger log-likelihood of training data may not necessarily lead to a lower testing error, particularly when the model is not fully trained.

4. Conclusions

In this paper, we propose a novel iterative scaling algorithm, named 'FIS'. Compared to the previous work on iterative scaling method, the FIS algorithm uses an auxiliary function that is able to bound the original log-likelihood function tighter. In our empirical studies of text classification problems over three datasets, the FIS method is able to converge significantly faster than the IIS algorithm and the CG algorithm. Furthermore, the new algorithm FIS is able to obtain 'decent' estimation of parameters (e.g. parameters resulting in low testing error) even when the learning process is still far away from the convergence point. As a future work, we would like to examine the effectiveness of this new iterative scaling algorithm on other tasks such as part of speech tagging.

References

Abramowitz, M. and Stegun, C. A. (Eds.) (1972) Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing. New York: Dover, p. 11, 1972.

D. Beeferman, A. Berger and J. Lafferty (1999), Statistical Models for Text Segmentation. In Machine Learning, 34:177-210, 1999.

A. Berger, V. Pietra and S. Pietra (1996), A Maximum Entropy Approach to Natural Language Processing. In Computational Linguistics, 22:39--71, 1996.

A. Berger (1997), The improved iterative scaling algorithm: A gentle introduction, www.cs.cmu.edu/afs/~aberger/www/ps/scaling.ps

A. Borthwick, J. Sterling, E. Agichtein and R. Grishman (1998), Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In Proceedings of the Sixth Workshop on Very Large Corpora, 1998.

J. Darroch and D. Ratcli (1972), Generalized iterative scaling for log-linear models. Annals of Math. Statistics, 43(5):1470-1480, 1972.

F. Jelinek (1997). Statistical Methods for Speech Recognition. The MIT Press, Cambridge, Massachusetts, London, England, 1997.

T. Minka (2001), Algorithms for maximum-likelihood logistic regression, CMU Statistics Tech Report 758, http://www.stat.cmu.edu/~minka/papers/logreg.html, 2001.

K. Nigam, J. Lafferty and A. McCallum (1999), Using Maximum Entropy for Text Classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999.

A. Ratnaparkhi (1996), A Maximum Entropy Model for Part-of-Speech Tagging. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1996.

R. Rosenfeld (1996), A Maximum Entropy Approach to Adaptive Statistical Language Modeling. In Computer, Speech and Language, 10:187-228, 1996.

J. Shewchuk(1994), An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, 1994.

M. Moller (1993), A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning, Neural Network, 6, 525-533.

R. Malouf (2002), A Comparison of Algorithms for Maximum Entropy Parameter Estimation, Proceedings of CoNLL-2002

D.C. Liu & J. Nocedal (1989) On the Limited Memory BFGS Method for Large Scale Optimization, Math. Prog. 45, 503-528, 1989

Appendix: Proof of the Extension of Holder Sum Inequality

The extension of Holder Sum Inequality claims that the following inequality will always hold

$$\sum_i \prod_k \alpha_{i,k} \le \prod_k \left\{ \sum_i \alpha_{i,k}^{q_k} \right\}^{1/q_k}$$

for any set of $\{q_k\}$, as long as all the q_k are positive and satisfies the constraint $\sum_k q_k^{-1} = 1$. The above inequality can simply be proved by the induction on k: (1) k=1: inequality in (6) holds because the RHS of the inequality is identical to the LHS of the inequality.

(2) Assuming inequality in (6) holds for any $k \le l$ and need to prove when $k=l+1$. Using Holder Sum Inequality, we have the following inequality hold

$$\begin{aligned} \sum_i \prod_{k=1}^{l+1} \alpha_{i,k} &\le \sum_i \alpha_{i,l+1} \left(\prod_{k=1}^{l} \alpha_{i,k} \right) \\ &\le \left(\sum_i \alpha_{i,l+1}^{p} \right)^{1/p} \left(\sum_i \left\{ \prod_{k=1}^{l} \alpha_{i,k} \right\}^{q} \right)^{1/q} \end{aligned} \tag{A1}$$

for any $p>1$, $q>1$ and $p^{-1} + q^{-1} = 1$. Now by letting $q=q_{l+1}$ and $p=1/(\Sigma_{k=1}^{l} q_k)$, we will have inequality (A1) further expanded as:

$$\begin{aligned} \sum_i \prod_{k=1}^{l+1} \alpha_{i,k} &\le \left(\sum_i \alpha_{i,l+1}^{p} \right)^{1/p} \left(\sum_i \left\{ \prod_{k=1}^{l} \alpha_{i,k} \right\}^{q} \right)^{1/q} \\ &\le \left(\sum_i \alpha_{i,l+1}^{q_{l+1}} \right)^{1/q_{l+1}} \left(\sum_i \left\{ \prod_{k=1}^{l} \alpha_{i,k} \right\}^{1/\sum_{j=1}^{l} q_j^{-1}} \right)^{\sum_{j=1}^{l} q_j^{-1}} \\ &= \left(\sum_i \alpha_{i,l+1}^{q_{l+1}} \right)^{1/q_{l+1}} \left(\sum_i \prod_{k=1}^{l} \alpha_{i,k}^{1/\sum_{j=1}^{l} q_j^{-1}} \right)^{\sum_{j=1}^{l} q_j^{-1}} \end{aligned} \tag{A2}$$

According to the induction assumption, the extension of Holder Sum Inequality holds for any $k \le l$. Therefore, we can use it to upper bound the second item in the RHS of above equation, i.e.

$$\begin{aligned} &\left(\sum_i \prod_{k=1}^{l} \alpha_{i,k}^{1/\sum_{j=1}^{l} q_j^{-1}} \right)^{\sum_{j=1}^{l} q_j^{-1}} \\ &\le \left\{ \prod_{k=1}^{l} \left(\sum_i \left(\alpha_{i,k}^{1/\sum_{j=1}^{l} q_j^{-1}} \right)^{q_i \sum_{j=1}^{l} q_j^{-1}} \right)^{\frac{1}{q_i \sum_{j=1}^{l} q_j^{-1}}} \right\}^{\sum_{j=1}^{l} q_j^{-1}} \\ &= \prod_{k=1}^{l} \left(\sum_i \alpha_{i,k}^{q_i} \right)^{q_i^{-1}} \end{aligned} \tag{A3}$$

By merging inequality (A3) and (A2) together, we have extension of Holder Sum Inequality proved when $k=l+1$. With this induction step, we proved the extension of Holder Sum Inequality is true for any k.