AUTOMATIC SPEECH RECOGNITION BASED ON WEIGHTED MINIMUM CLASSIFICATION ERROR (W-MCE) TRAINING METHOD

Qiang Fu, Biing-Hwang Juang

{qfu,juang}@ece.gatech.edu
School of Electrical & Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332

ABSTRACT

The Bayes decision theory [1] is the foundation of the classical statistical pattern recognition approach. For most of pattern recognition problems, the Bayes decision theory is employed assuming that the system performance metric is defined as the simple error counting, which assigns identical cost to each recognition error. However, this prevalent performance metric is not desirable in many practical applications. For example, the cost of "recognition" error is required to be differentiated in keyword spotting systems. In this paper, we propose an extended framework for the speech recognition problem with non-uniform classification/recognition error cost. As the system performance metric, the recognition error is weighted based on the task objective. The Bayes decision theory is employed according to this performance metric and the decision rule with a non-uniform error cost function is derived. We argue that the minimum classification error (MCE) method, after appropriate generalization, is the most suitable training algorithm for the "optimal" classifier design to minimize the weighted error rate. We formulate the weighted MCE (W-MCE) algorithm based on the conventional MCE infrastructure by integrating the error cost and the recognition error count into one objective function. In the context of automatic speech recognition (ASR), we present a variety of training scenarios and weighting strategies under this extended framework. The experimental demonstration for large vocabulary continuous speech recognition is provided to support the effectiveness of our approach.

Index Terms: non-uniform error cost, weighted MCE

1. INTRODUCTION

The Bayes decision theory [1] is the foundation of statistical pattern recognition. This theory can be summarized as follows. First, the performance of a pattern recognizer(i.e., how good the recognizer's decision of class identity is) is to be measured in statistical terms, meaning that we are interested in the system's *expected* performance, not the performance towards any particular pattern. Second, the recognizer's decision or recognition policy has to be formed based on full knowledge of the probability distribution of each and every class. Third, the recognition decisions are made so that the *expected* loss over the entire data set is minimized. Let us assume that for an arbitrary observation X, a conditional loss for classifying X into a class i event can be defined as [1]

$$R(C_i|X) = \sum_{j=1}^{M} e_{ij} P(C_j|X) \tag{1}$$

where $P(C_j|X)$ is the *a posteriori* probability. The error cost function can be denoted as an $M \times M$ matrix with entries e_{ij} where $i, j \in I_M$, signifying the cost in identifying a pattern from the j^{th} class as one of the i^{th} class. Note that if $e_{ij} = 0$, for $i = j$ and $e_{ij} = 1$, for $i \neq j$, the cost function is called uniform and leads to simple error counting, which is one of the most intuitive and prevalent performance measures in pattern recognition. Hence, the expected loss function is written as

$$\mathcal{L} = \int R(C(X)|X) p(X) dX \tag{2}$$

where $C(X)$ is the decision function. It is obvious that if for every X, the classifier satisfies

$$R(C(X)|X) = \min_i R(C_i|X) \tag{3}$$

the expected loss in (2) will be minimized. If we impose the assumption that the error loss function is uniform (i.e., $e_{ij} = 0$, for $i = j$ and $e_{ij} = 1$, for $i \neq j$), the conditional loss becomes

$$R(C_i|X) = \sum_{i \neq j} P(C_j|X) = 1 - P(C_i|X) \tag{4}$$

This leads to the *maximum a posteriori* (MAP) decision rule [1], in which the decision function is defined as

$$C(X) = i \quad \text{if} \quad P(C_i|X) = \max_j P(C_j|X) \tag{5}$$

The minimum error rate achieved by the MAP rule is called the *Bayes risk*. Based on (5), the minimum error rate can be achieved if we can estimate $P(C_j|X)$ precisely, which transforms the classifier design problem into a distribution estimation problem. The maximum likelihood (ML) training criterion [1] and the expectation-maximization (EM) algorithm [1] are usually employed in the distribution estimation.

The classical pattern recognition approach has been applied to many practical applications such as automatic speech recognition (ASR). However, its optimality is often compromised due to several substantial limitations rooted in the fundamental assumptions. One important issue is that the MAP rule can not be effectively implemented due to lack of full knowledge of the data distribution. Further more, the data size is normally insufficient for reliable parameter estimation even when the correct distribution form is given. Hence, the Bayes risk is generally unachievable and distribution estimation can rarely lead to the optimal classifier design. This fact motivated the minimum classification error (MCE) training method [2] [3] which

aims at minimizing the empirical error rate directly. The other limitation of the classical pattern recognition approach comes from the fundamental assumption of the performance metric. By default, the error cost function in the classical performance metric is uniform; i.e., $e_{ij} = 0$, for $i = j$ and $e_{ij} = 1$, for $i \neq j$. However, to measure a recognizer's performance, the assumption of identical error cost is not always reasonable in practice. In a variety of ASR applications, some errors should be considered more critical than others in terms of the system objective. For example, a keyword spotting system may consider a recognition error of a "key" word unacceptable, while errors of functional words such as "a" or "the" may not be considered consequential. Another example is the composite system of speech recognition and understanding. The speech understanding system does not need a perfect transcription of the whole utterance. The performance is affected seriously by the correctness of some essential words/phones. In these cases, the differentiation of the significance of the recognition error is necessary and a nonuniform error cost function becomes appropriate.

With the change in the performance metric to a non-uniform cost, the use of the Bayes decision theory needs to be revised accordingly. The objective of the classifier design is still to minimize the expected loss in (2) via Eq.(3). However, since e_{ij} is no longer uniform, the MAP rule is not applicable and the decision rule is restated as

$$C(X) = C_i = \arg\min_i \sum_{j=1}^{M} e_{ij} P(C_j|X) \tag{6}$$

From (6), we see that the right decisions are made to minimize a weighted combination of the error probabilities instead of simply maximizing the *a posteriori* probability. If we revisit the rationale of the formulation of pattern recognition problems, the conditional loss of (1) which consists of the class-dependent error loss function and the *a posteriori* probability is the root of the derivation of any specific decision rule. The MAP rule imposes a strong constraint of uniform error loss function and highly relies on the precise estimation of the *a posteriori* probability. The decision rule of (6), on the contrary, relaxes the constraint of the error loss function and transforms the classifier design into an error cost minimization problem instead of a distribution estimation problem. Therefore, a training algorithm that can minimize the weighted error rate effectively is needed.

As pointed out by [2], the distribution estimation methods can not really achieve the MAP rule due to lack of knowledge of the distribution forms, not to mention to be suitable for the generalized decision rule of (6). The discriminative training (DT) methods [4] recently arose as an important family of alternative training methods, especially for speech recognition problems. Conventionally, there are three types of popular training algorithms in ASR. They are: the *maximum mutual information* (MMI) [5], the *minimum phone/word error* (MPE/MWE) [6] and the *minimum classification error* (MCE) method [2][3]. The MMI algorithm aims at maximizing the mutual information between the data observation distribution and the distribution of the corresponding label. Though being employed in many practical applications, it does not follow the MAP rule and is not aiming at the minimum error rate. The MMI method is also not applicable for the optimal classifier design under the generalized decision rule with non-uniform loss functions. The MCE method is a suitable training algorithm for the decision rule of (6) after appropriate generalization. First, the objective function of the MCE method is constructed to approximate the recognition error explicitly on a token-by-token basis. It is easy to integrate a non-uniform error cost function or other weighting functions into the MCE objective function since each training toke has a known class label. Second, the MCE method computes the contribution of each training token to the expected loss over the entire training set (if uniform error loss is assumed, the expected loss equals the expected error rate), maintaining the use of an empirical estimate of the system performance even when error weighting is included. We will see in this paper later that the MPE/MWE method could be viewed as a special version of the weighted MCE method.

In this paper, we propose an extended pattern recognition framework in which the performance metric is generalized to non-uniform cost functions. The *minimum risk* (MR) decision rule and its practical implementation strategy are constructed according to the error weighting mechanism in the performance metric. Based on the generalized performance metric and the extended decision rule, an effective training algorithm called weighted MCE (W-MCE) is developed for minimizing the weighted error rate. Two important training scenarios and corresponding error weighting mechanisms are discussed in the context of ASR applications.

The rest of paper is organized as follows. In the next section, we provide a justification example of the non-uniform error rate as the performance metric. The non-uniform error cost framework and the weighted MCE method are introduced in Section 3. In Section 4, we summarize a variety of weighting mechanisms under different training scenarios in ASR. In addition, the relationship between the weighted MCE method and the minimum phone/word error (MPE/MWE) method is discussed. In Section 5, we report some experimental results to demonstrate the effectiveness of the non-uniform error cost and the W-MCE method. A comprehensive conclusion and discussion of future work are finally presented in Section 6.

2. AN EXAMPLE FOR NON-UNIFORM ERROR RATE

Here is an example for using the non-uniform error rate as the recognition performance measure for better information understanding. Two recognized strings with an identical equal-significance word error rate are displayed as follows:

0 AT N. E. C. THE NEED FOR INTERNATIONAL MANAGERS WILL KEEP RISING

1 AT *ANY* $<$ *del* $>$ *SEE* THE NEED FOR INTERNATIONAL MANAGERS WILL KEEP RISING

2 AT N. E. C. $<$ *del* $>$ *NEEDS* FOR INTERNATIONAL *MANAGER'S* WILL KEEP RISING

Item 0 is a transcription of the first utterance (440c0201.lab) in the test set of the Wall Street Journal (WSJ0) [7] database. Item 1 displays a recognition result with two substitution errors and one deletion error. Item 2 is another recognition result with the same error counts. These two recognition results contributed to identical error statistics in terms of the equal-significance word error rate. However, we can retrieve the correct information from the second string with almost no difficulty but the company name is totally lost in the first one. Hence, the second string should be viewed as a better recognition result because it has retained useful information.

Consider the task of acoustic modeling for words and the differentiation in error significance is being applied to words. One straightforward error significance weighting function in this scenario is the Shannon information of word $-\log P(word)$ in the whole training corpus. This weighting function reasonably assumes that

the less frequently a word appears, the more information it contains. The weighted word error rate (WWER) can thus be calculated as

$$\text{WWER} = \frac{-[\sum_{s=1}^{S} \log P(w_s) + \sum_{d=1}^{D} \log P(w_d) + \sum_{i=1}^{I} \log P(w_i)]}{\sum_{n=1}^{N} [-\log P(w_n)]} \tag{7}$$

where N is the total number of words, and S, D, and I are the number of substitution, deletion and insertion errors, respectively. w_s,w_d, and w_i are the words in the corresponding errors in substitution, deletion and insertion, respectively. We obtain the following table in which the Shannon information $-\log P(word)$ of each word is listed below for each word sequence:

0 AT N. E. C. THE NEED FOR INTERNATIONAL MANAGERS WILL KEEP RISING

2.317 3.138 3.135 2.784 1.275 3.675 2.027 3.259 3.797 2.481 3.689 3.925

1 AT *ANY* $<$ *del* $>$ *SEE* THE NEED FOR INTERNATIONAL MANAGERS WILL KEEP RISING

2.317 3.038 $<$ *del* $>$ 3.503 1.275 3.675 2.027 3.259 3.797 2.481 3.689 3.925

2 AT N. E. C. $<$ *del* $>$ *NEEDS* FOR INTERNATIONAL *MANAGER'S* WILL KEEP RISING

2.317 3.138 3.135 2.784 $<$ *del* $>$ 3.966 2.027 3.259 3.719 2.481 3.689 3.925

Based on Eq.(7), the weighted recognition error rate of the first string is 27.25% while the second one outperforms this number and achieves 25.24%. This example demonstrates the effectiveness of the weighted word error rate. Other significance weighting functions are possible. For example, one could associate proper nouns with substantially higher significance than common words. The error rate differentiation could have been much higher than about 2% as demonstrated.

3. THE MINIMUM RISK DECISION RULE AND WEIGHTED MCE METHOD

3.1. The Minimum Risk Decision Rule

Based on the Bayes decision theory, to minimize the expected loss function $\mathcal{L}$ defined in (2), the minimum conditional loss $R(C_i|X)$ of (1) is wanted for each decision. For a general error cost function e_{ij}, the classifier $C(X)$ that satisfies (1) and (2) can be written explicitly as

$$C(X) = C_i = \arg\min_i \sum_{j=1}^{M} e_{ij} P(C_j|X) \tag{8}$$

$$= \arg\min_i \sum_{j=1}^{M} e_{ij} P(X|C_j) P(C_j)/P(X) \tag{9}$$

We name this decision rule as the *minimum risk* (MR) rule. In practice, we generally require that $e_{ij} = 0$ for $i = j$ and $e_{ij} \geq 0$ for $i \neq j$. The MR rule of (9) does not lead to the MAP policy of (5) even if the knowledge of the true distribution (*a posteriori* probabilities) is available to the recognizer. Implementation of the MR rule requires multiplication if the cost matrix and the posterior probability vector, a direct result of the non-uniformity of the cost function.

Execution of (9) obviously requires the knowledge of the *a posteriori* probability $P(C_j|X), \forall j \in I_M$. As stated in [2], however, the true *a posteriori* probability is rarely available for a number of reasons (e.g., lack of knowledge of the distribution forms or sufficient labeled data for accurate estimation of the distribution parameters). Any decision rule such as the MAP policy that requires precise knowledge of the *a posteriori* probability cannot be accomplished in practice. The decision rule of (9), which involves a weighted combination of the *a posteriori* probabilities for all the classes, may demand even more crucially the availability of the *a posteriori* probability than the MAP policy (which in effect only requires that the rank order of the *a posteriori* probabilities be accurately preserved in the evaluation.) Furthermore, the complexity of the conditional cost of (9) may impose additional difficulties for the system designer to associate appropriate training models with the given data in the optimization process.

3.2. A Practical MR Rule and The Weighted MCE Method

To overcome the implementation difficulties of the MR rule in system training, the expected system loss of (2) needs to be expressed in terms of the empirical loss (yet to be defined) with a more practical decision rule embedded in it. For clarity, let $i_X = C(X)$ be the identity index as decided by the recognizer and j_X be the true identity index of X. Also, $\Omega = \{X_k\}_{k=1}^{K}$ is the set of training tokens. Then, in practice, the expected loss $\mathcal{L}$ can be approximated as an alternative non-uniform risk, which is an accumulation of the actual single token empirical cost:

$$\mathcal{L} = \int \sum_{j=1}^{M} e_{i_X j} P(C_j|X) p(X) dX \tag{10}$$

$$\Uparrow$$

$$L = \frac{1}{K} \sum_{x \in \Omega} l_{i_X j_X}(X_k) = \frac{1}{K} \sum_{x \in \Omega} e_{i_X j_X} \tag{11}$$

in which $l_{i_X j_X}(X_k) = e_{i_X j_X}$ is the error cost for training token $X_k \in j_X$ being classified into class i_X. Therefore if the empirical system loss is defined over the token-based costs, to evaluate (11), one can prescribe a discriminant function for each class, $g_j(X;\Lambda), \forall j$, and define the practical decision rule for the recognizer as

$$C(X) = i_X = \arg\max_i g_i(X;\Lambda) \tag{12}$$

The alternative system loss is then

$$L' = \sum_{X \in \Omega} \sum_{i \in I_M} \sum_{j \in I_M} e_{ij} \mathbf{1}[X \in C_j] \mathbf{1}\{i = \arg\max_m g_m(X;\Lambda)\} \tag{13}$$

Therefore, Eq.(13) can be rewritten as

$$L' = \sum_{j \in I_M} L_j \tag{14}$$

and

$$L_j = \sum_{X \in \Omega} \Big(\sum_{i \in I_M} e_{ij} \mathbf{1}\{i = \arg\max_m g_m(X;\Lambda)\} \Big) \mathbf{1}[X \in C_j] \tag{15}$$

That is, L_j is the empirical error cost collected over all training tokens in Ω with $j_X = j$. The approximation then needs to be made

to the summands. This can be accomplished by

$$\sum_{i\in I_M} e_{ij}\mathbf{1}\{i=\arg\max_m g_m(X;\Lambda)\} \approx \sum_{i\in I_M} \frac{e_{ij}g_i^\eta(X;\Lambda)}{g_i^\eta(X;\Lambda)+G_i(X;\Lambda)} \tag{16}$$

where

$$G_i(X;\Lambda) = \sum_{m\in I_M, m\neq i} g_m^\eta(X;\Lambda) \tag{17}$$

Note that as $\eta\to\infty$,

$$\frac{g_i^\eta(X;\Lambda)}{g_i^\eta(X;\Lambda)+G_i(X;\Lambda)} \approx \begin{cases} 1, & g_i(X;\Lambda)=\max_m g_m(X;\Lambda) \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

Finally, the objective function of the weighted MCE (i.e., the smoothed empirical system cost) is

$$L' \approx \sum_{X\in\Omega}\sum_{j\in I_M}\sum_{i\in I_M} e_{ij}\frac{g_i^\eta(X;\Lambda)}{g_i^\eta(X;\Lambda)+G_i(X;\Lambda)}\mathbf{1}[X\in C_j] \tag{19}$$

which can be rewritten as

$$L' \approx \sum_X\sum_j\sum_i e_{ij}\frac{1}{1+\exp\{-\ln g_i^\eta+\ln G_i\}}\mathbf{1}[X\in C_j] \tag{20}$$

which is a continuous function of the parameter set Λ. Similarly, the hyper-parameter η can be chosen tradeoff between approximation and smoothness. In speech recognition applications, if we let $g_i(X;\Lambda) = P(X|C_i)$ and $P(X|C_i)$ is constructed as a hidden Markov model (HMM) for class i, we can use the gradient descent methods (e.g., generalized probabilistic descent method (GPD)[2]) to optimize parameters in (20). A similar optimization algorithm as the one in [2] can be derived for (20).

4. TRAINING SCENARIOS AND WEIGHTING STRATEGIES IN ASR

Speech recognition is an important category of the pattern recognition applications with many different training and recognition scenarios. In brief, there are two major non-uniform error cost training scenarios in ASR. The first scenario is that the training and recognition decisions are on the same sematic level with the performance measure. For example, the acoustic model is trained on the *phone* level and the evaluation metric is the weighted *phone* error rate (PER). In this case, the loss of the wrong recognition decisions represents the recognizer's performance directly. We call this scenario the **intra-level training**. The second and the most common circumstance in practice is the **inter-level** training in which the training and recognition decisions are on the different semantic level with the performance metric. For example, the training and the recognition are on the *phone* level but the system evaluation measure is the *word* error rate (WER). In this case, the system performance is not instantly evaluated by the recognition error loss. Hence, minimizing the cost of the wrong recognition decisions does not directly optimize the recognizer's performance in terms of the evaluation metric. To alleviate this inconsistency, the error weighting strategy could be built in a cross-level fashion.

In both training scenarios, the error weighting mechanism can be built according to two types of error cost: the user-defined cost and the data-defined cost. The user-defined cost is usually characterized by the system requirement and relatively straightforward. On the other hand, the data-defined cost is more complicated. The Bayes decision theory aims at minimizing the expected error loss, which is generated by the wrong recognition decisions. The wrong decisions occur because the underlying data observation deviates from the distribution represented by the corresponding recognizer model. Though we can not distinguish whether a decision error is due to "bad" data or "bad" models, it is possible to measure the "reliability" of the errors by introducing the data-defined weighting. Through data-defined weighting, the recognizer modeling would bias more to the "good" data.

4.1. Error Weighting for Intra-Level Training

In the intra-level training situation, the system performance is directly measured by the loss of wrong recognition decisions. Hence, we can absorb both types of the error weighting into the error cost function e_{ij} as one universal functional form. Assume that the training is on the *phone* level and the evaluation measure is the weighted *phone* error rate (PER). We employ the phone sequence $PH = (ph_1, ph_2, \ldots, ph_{L_k})$ to represent the label of the kth training token in a training set with totally K tokens. $X_k = \{X_{k,l_k}\}_{l_k}^{L_k}$ is the kth token that is segmented into L_k segments corresponding to the phone sequence. Recall the decision rule (12), we define $g_i(X;\Lambda) = P(X;\Lambda|C_j)P(C_j)$. Finally, following (20), the objective function for the weighted MCE in this case could be written as

$$\mathcal{F}_{W-MCE}^{(1)} = \sum_k\sum_{l_k=1}^{L_k}\sum_i l_i(X_{k,l_k};\Lambda)e_{ij}\mathbf{1}[X_{k,l_k}\in C_j] \tag{21}$$

where

$$l_i(X_{k,l_k};\Lambda) = \frac{1}{1+\exp\{-\ln g_i^\eta(X_{k,l_k};\Lambda)+\ln G_i(X_{k,l_k};\Lambda)\}} \tag{22}$$

and $G_i(X_{k,l_k};\Lambda) = \sum_{m\in I_M, m\neq i} g_m^\eta(X_{k,l_k};\Lambda)$.

4.2. Error Weighting for Inter-Level Training

In the inter-level training situation, the system performance is not measured directly by the loss of the wrong recognition decisions. The recognition decisions need to be grouped to form the system output which is on the level of the performance metric. Therefore, we need to use cross-level weighting in this case to break down the high level cost and impose the appropriate weights upon the low level models.

Assume that in this case, the training is on the phone model and the performance metric is the weighted word error rate. The first weighting mechanism we are discussing is the user-defined weighting. Let the word sequence $W = (w_1, w_2, \ldots, w_{L_k})$ be the label of the kth training token in a training set with totally K tokens. Each word w_{l_k} contains a phone sequence as $ph_{l_k}^1, \ldots, ph_{l_k}^{n_k}, \ldots, ph_{l_k}^{N_k}$. $X_k = \{X_{k,l_k,n_k}\}_{l_k}^{L_k}$ is the kth token that is segmented into L_k segments corresponding to the word sequence. Since the user's demands are normally engaged on the level of the system performance metric, the user-defined cost of each phone in word w_{l_k} could be set identically using the word-level cost. Hence, the user-defined weighting of the weighted MCE in the inter-level training can be written as:

$$\mathcal{F}_{W-MCE}^{(2)} = \sum_k\sum_{l_k=1}^{L_k}\sum_i l_i(X_{k,l_k};\Lambda)\mathbf{1}[X_{k,l_k}\in C_j]e_{ij}^{(u)}(w_{l_k}) \tag{23}$$

where $e_{ij}^{(u)}(w_{l_k})$ is the user-defined cost for word w_{l_k}. One instance of the user-defined error weighting function is the Shannon information $e_{ij}^{(u)}(w_{l_k}) = -\log P(w_{l_k})$ as we mentioned in Section 2.

The formulation of the data-defined weighting is more complex and flexible in the inter-level training. Since the objective of the data-defined weighting is to find the "reliable" errors, the data-defined weighting can be imposed upon any semantic levels. In this situation, the objective function of the weighted MCE method can be written as follows:

$$\mathcal{F}_{W-MCE}^{(2)} = \sum_k \sum_{l_k=1}^{L_k} \sum_i l_i(X_{k,l_k};\Lambda)\mathbf{1}[X_{k,l_k} \in C_j] e_{ij}^{(d)}(m) \tag{24}$$

where $m = \{n_k, l_k, k\}$. $e_{ij}^{(d)}(m)$ could be the data-defined weighting for the n_k^{th} phone, the l_k^{th} word, or the k^{th} training token. The definition of the data-defined error is very flexible. We can assign the cost of the l_k^{th} word to each phone inside or compute the phone-level error cost for phone n_k in word l_k separately. One example of the data-defined weighting is the popular MPE/MWE method.

A W-MCE objective function including both weighting functions under the inter-level training scenario can be written as

$$\mathcal{F}_{W-MCE}^{(2)} = \sum_k \sum_{l_k=1}^{L_k} \sum_i l_i(X_{k,l_k};\Lambda)\mathbf{1}[X_{k,l_k} \in C_j] e_{ij}(w_{l_k}, m) \tag{25}$$

where $m = \{n_k, l_k, k\}$ and $e_{ij}(w_{l_k}, m)$ is the total weights. The definition of l_i in (23), (24) and (25) are identical as of Eq.(22).

4.3. Weighted MCE and MPE/MWE Method

In this section, we will discuss the relation between the weighted MCE and the MPE/MWE method in order to contribute a better understanding of the error weighting mechanism. Assume that the training is on the phone model and the performance metric is the word error rate. We use $W = (w_1, w_2, \ldots, w_{L_k})$ to denote the label of the kth training utterance in a training set with totally K tokens. $X_k = \{X_{k,l_k}\}_{l_k}^{L_k}$ is the kth token that is segmented into L_k segments corresponding to word/phone sequence. In this section, we assume that there is no user-defined weighting for simplicity. As we discussed before, the objective function of the weighted MCE method is defined as of (24).

The minimum phone/word error (MPE/MWE) training method is a popular discriminative training method with a weighted objective function to mimic training errors [6]. The objective function of MPE/MWE is defined as follows:

$$\begin{aligned}
\mathcal{F}_{MPE/MWE} &\approx \sum_{k=1}^{K} \frac{\sum_{l_k \in L_k} p^{\alpha}(X_k|w_{l_k}) P^{\beta}(w_{l_k}) A(W, W_k)}{\sum_{\forall u} p^{\alpha}(X_k|w_u) P^{\beta}(w_u)} \\
&= \sum_{k=1}^{K} \frac{P_c A(W, W_k)}{P_c + P_w} \\
&\approx \sum_{k=1}^{K} [1 - \sum_{u \neq l_k} l'_u(X_{k,l_k};\Lambda)] A(W, W_k)
\end{aligned} \tag{26}$$

where

$$\sum_{u \neq l_k} l'_u(X_{k,l_k};\Lambda) = \frac{P_w}{P_c + P_w} = \frac{1}{1 + \exp\{-\ln P_w + \ln P_c\}} \tag{27}$$

and

$$P_c = \sum_{l_k \in L_k} p^{\alpha}(X_{k,l_k};\Lambda|w_{l_k}) P^{\beta}(w_{l_k}) \tag{28}$$

$$P_w = \sum_{u \neq l_k} p^{\alpha}(X_{k,l_k};\Lambda|w_u) P^{\beta}(w_u) \tag{29}$$

In Eq.(26), $A(W, W_k)$ is called "raw accuracy", which is a measure of how many words/phones are correctly recognized in W_k according to the transcription W [6]. We may interpret it as a rough estimation of the word/phone error accumulation for each utterance. α and β are the acoustic and language model scale factors. We use P_c to represent the probability of applying the training observations on the transcribed string, and P_w as the sum of applying the observations on all the other recognized strings.

Therefore, to maximize the original MPE/MWE objective function in (26) is equivalent to minimize the modified objective function below, which is very similar to the objective function of the W-MCE method defined in (24).

$$\mathcal{F}'_{MPE/MWE} = \sum_{k=1}^{K} \sum_i l'_i(X_{k,l_k};\Lambda) A(W, W_k) \tag{30}$$

In summary,the MPE/MWE method weights the utterance errors by the "raw accuracy" $A(W, W_k)$, therefore builds a objective function that incorporates the non-uniform error cost of each training utterance. Hence, the relationship between the weighted MCE and the MPE/MWE can be described as two training algorithms both rooted in the Bayes decision theory, directing to the same aim of designing the optimal classifier to minimize the non-uniform error cost.

5. EXPERIMENT RESULTS FOR WEIGHTED MCE

In this paper, we employ the weighted MCE method on large vocabulary continuous speech recognition tasks. The weighing function is an estimate of the "posterior probability" of the target speech unit w_{l_k} given the data observations of the corresponding training utterance X_1^T. Note that the term "posterior probability" does not obey the rigorous mathematical definition but only a reasonable engineering approximation.

5.1. Baseline

The experiments are carried out on the WSJ0 database [7].The baseline recognizer follows the recipe for WSJ database using the HTK toolbox (http://www.inference.phy.cam.ac.uk/kv227/htk/), which is based on representing training classes using continuous density Gaussian mixture hidden Markov models (CDHMM). A word internal context-dependent tri-phone set is formed with 7,385 physical models and 19,075 logic models. All models are represented by 3-state strict left-to-right HMMs, with 8 Gaussian mixture components per state. These models were initialized by the Baum-Welch method [8]. The experiments were then carried out by comparing the performance of systems trained using different MCE criterion.

We generate feature vectors for all 7,077 utterances by 84 speakers in the training set of the WSJ0 corpus. Each feature vector has 12MFCC+12Δ+12Δ^2 and 3 log energy values so that total 39 features are used. The feature generation process is also applied on the Nov-92 evaluation set with 330 utterances by 8 speakers. The CMU6 recognition lexicon are employed, which contains 126,834 words. We conduct the similar word-graph training implementation as [9]. During the training procedure, a unigram language model is used. Bigram is applied to decode and generate word graphs, where

the word insertion penalty and the language model scale factor are set to be -4.0 and 15.0, respectively. At most 3 candidate recognition strings are allowed to survive simultaneously during word graph generation. Other baseline system details can be found in [10]. Finally, the word error rate of this baseline is 8.41%.

5.2. W-MCE Implementation and Experiments

In our experiments, we assume that the weighting function only contains the data-defined weighting for simplicity. The objective function of the weighted MCE in our experiment can be written as

$$\mathcal{F}_{W-MCE} = \frac{1}{K}\sum_{k=1}^{K}\sum_{l_k=1}^{L_k}\sum_{i} l_i(X;\Lambda)Pr(w_{l_k}|X_1^T) \qquad (31)$$

where

$$Pr(w_{l_k}|X_1^T) = \sum_{\forall n, w_n = w_{l_k}} \frac{P^{\alpha}(X_1^T|w_{l_k})P^{\beta}(w_{l_k}|w_{l_k-1})}{P^{\alpha}(X_1^T)}$$
$$\text{when} \quad |s_n - s| + |t_n - t| < \tau \qquad (32)$$

$[s_n, t_n]$ and $[s, t]$ are the starting and ending time of w_n and w_{l_k}, respectively. τ is the parameter to control boundary relaxation. The GPD method [2] is employed to minimize (31) and we assume that the weighting function $Pr(w_{l_k}|X_1^T)$ is fixed during each optimization epoch.

We compared the weighted MCE and the conventional MCE method in terms of the word error rate respectively on three semantic levels: the word level, the phone level and the state level [9]. The word level training means the loss function $l_i(X_{l_k};\Lambda)$ is computed for each word w_{l_k}. Similarly, the phone level training means that the error cost is computed on per phone basis, and so does the state level training.

In Table 1, we can see that the W-MCE method outperforms the MCE method in all categories. In both methods, the phone-level training achieves better performance than the word level and state level training. The reason for this observation may be that the time interval for state level training to calculate the error loss is too short, and the time interval for word level is too long. Too short interval could lead to over-optimization. Too long interval contains too many parameters so that the effect for each parameter is weakened when maximizing the corresponding objective.

Table 1. Word Error Rate (WER) for WSJ0-eval using the MCE method and the W-MCE method

Training level	MCE	W-MCE
Baseline	8.41	8.41
Word-level	8.16	8.11
Phone-level	7.96	7.73
State-level	8.11	8.05

6. CONCLUSION AND FUTURE WORK

In this paper, we address one common misinterpretation of the classical Bayes decision theory and introduce a generalized non-uniform error cost framework for automatic speech recognition. After a justification ASR example of using the non-uniform error rate as the performance metric, We propose the minimum risk (MR) decision rule. A practical alternative decision rule is provided due to some implementation issues for the original MR rule. We then present the formulation of the weighted minimum classification error (W-MCE) algorithm.

Two practical speech recognition training scenarios and the corresponding error weighting schemes are discussed. In addition, we discuss the relationship between the weighted MCE method and the minimum phone/word error (MPE/MWE) method. We provide some demonstration experiment results to support the effectiveness of the non-uniform error criteria and the weighted MCE method.

In the future, we plan to build a complete framework of the non-uniform error criteria for speech recognition and even further, the general pattern recognition problem. As a rich research area, there are many promising open topics such as the weighting techniques and optimization algorithms.

7. ACKNOWLEDGEMENTS

The authors would like to thank Antonio Moreno-Daniel for his help This work was supported in part by Microsoft research and AT&T.

8. REFERENCES

[1] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York, NY: John Wiley, 2001, 2nd edition.

[2] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 257–265, May 1997.

[3] W. Macherey, L. Haferkamp, R. Schluter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," in *Interspeech-2005*, Lisbon, Portugal, Sep. 2005, pp. 2133–2136.

[4] R. Schluter, W. Macherey, B. Muller, and H. Ney, "Comparison of discriminative training criteria and optimization methids for speech recognition," *Speech Communication*, vol. 34, no. 1, pp. 287–310, May. 2001.

[5] Y. Normandin, "Hidden markov models, maximum mutual infomation estimation, and the speech recognition problem," Ph.D. dissertation, McGill University, Montreal, Canada, 1991.

[6] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University, Cambridge, Britain, Jul. 2004.

[7] D. S. Pallet, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. A. Lund, and M. A. Przybocki, "1994 benchmark test for the arpa spoken language program," in *ARPA Human language Technology Workshop*, Austin, Texas, Jan. 1995, pp. 5–36.

[8] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[9] Q. Fu, A.Moreno, B. H. Juang, J.-L. Zhou, and F.Soong, "Generalization of the minimum classification error (mce) training based on maximizing generalized posterior probability (gpp)," in *ICSLP-2006*, Pittsburgh, PA, Sep. 2006.

[10] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, "Large vocabulary continuous speech recognition using htk," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Adelaide, Australia, Apr. 1994, pp. 125–128.