Available online at www.sciencedirect.com
ScienceDirect
Computer Speech and Language 24 (2010) 589–608
COMPUTER SPEECH AND LANGUAGE
www.elsevier.com/locate/csl

Discriminative training of HMMs for automatic speech recognition: A survey

Hui Jiang *

Department of Computer Science and Engineering, York University, 4700 Keele Street, Toronto, Ontario, Canada M3J 1P3

Received 18 February 2009; received in revised form 12 August 2009; accepted 13 August 2009
Available online 26 August 2009

Abstract

Recently, discriminative training (DT) methods have achieved tremendous progress in automatic speech recognition (ASR). In this survey article, all mainstream DT methods in speech recognition are reviewed from both theoretical and practical perspectives. From the theoretical aspect, many effective discriminative learning criteria in ASR are first introduced and then a unifying view is presented to elucidate the relationship among these popular DT criteria originally proposed from different viewpoints. Next, some key optimization methods used to optimize these criteria are summarized and their convergence properties are discussed. Moreover, as some recent advances, a novel discriminative learning framework is introduced as a general scheme to formulate discriminative training of HMMs for ASR, from which a variety of new DT methods can be developed. In addition, some important implementation issues regarding how to conduct DT for large vocabulary ASR are also discussed from a more practical aspect, such as efficient implementation of discriminative training on word graphs and effective optimization of complex DT objective functions in high-dimensionality space, and so on. Finally, this paper is summarized and concluded with some possible future research directions for this area. As a technical survey, all DT techniques and ideas are reviewed and discussed in this paper from high level without involving too much technical detail and experimental result.

1. Introduction

Over the past few decades, automatic speech recognition (ASR) has achieved huge success and its performance has been significantly improved in a variety of real-world applications, from simple digit recognition to large vocabulary broadcast news transcription, from reading style voice dictation to spontaneous dialogue systems, etc. These impressive advances are mainly attributed to many powerful statistical modeling techniques which have been broadly accepted in ASR for representing real data, such as speech signals and spoken language documents collected from real-world applications. As it is well known, the most successful modeling approach in ASR is to use hidden Markov models (HMMs) as acoustic models for sub-word (such as phonemes, syllables, etc.) or whole-word speech units and to use Markov chain model (a.k.a. *n*-gram model) as

* Tel.: +1 416 736 2100x33346; fax: +1 416 736 5872.
E-mail address: hj@cse.yorku.ca.

0885-2308/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2009.08.002

language model for sentences or word sequences in text documents. In these methods, it is essential to effectively learn all model parameters, including those of HMMs and *n*-gram models, from a large amount of training data according to certain training criteria. It has been shown that success of this data-driven modeling approach highly depends on the goodness of estimated models and the underlying modeling technique plays an critical role in the final system performance.

As we know, various statistical learning approaches have been extensively studied in the field of machine learning (Jebara, 2002; Vapnik, 1998). Generally speaking, there exist two distinct categories of learning algorithms in machine learning for building an effective pattern classifier, namely generative learning and discriminative learning. The generative learning scheme aims to estimate probability distribution of data for each class using density estimation methods. To make the estimation problem more feasible, the so-called parametric modeling approach (Jiang et al., 1999) has been widely adopted, where it is assumed that unknown probability distributions belong to some computationally tractable function families, such as the exponential family (Brown, 1986) or a finite mixture of exponential family distributions. In this way, the difficult density estimation problem turns into a more tractable parameter estimation problem. Conventionally, all unknown parameters of the presumed probability distributions are estimated from all available training samples according to the well-known maximum likelihood (ML) approach. At last, the estimated models are used for classification based on the Bayes decision rule (a.k.a. maximum *a posterior* decision rule) from statistical decision theory. It has been proven that the generative learning method leads to the optimal classifier as long as the presumed probability models indeed represent the true distribution of data (Nadas, 1983; Nadas et al., 1988). The major advantage of generative learning is that it is relatively easy to exploit inherent dependency or various relationship of data by imposing all kinds of structure constraints in generative learning, such as graphical models (Jordan, 2004). More importantly, many efficient learning algorithms, such as the Expectation–Maximization (EM) algorithm (Dempster et al., 1977; Neal and Hinton, 1998), are available for estimating a variety of generative models, even for many rather complicated models. As a result, the generative learning scheme has become a very popular data modeling approach for classification and regression in many practical applications. In ASR, the generative learning strategy has been extensively explored for estimating various types of HMMs, including discrete density HMMs (DDHMMs) and Gaussian mixture continuous density HMMs (CDHMMs), using the Baum–Welch (BW) algorithm (Baum et al., 1970), which is basically derived based on the Expectation–Maximization (EM) algorithm (Dempster et al., 1977) for maximum likelihood estimation (MLE) of HMMs.

On the other hand, the discriminative learning scheme has recently gained tremendous popularity in machine learning since it makes no explicit attempt to model the underlying distribution of data and instead it directly optimizes a mapping function from the input data samples to the desired output labels. Therefore, in discriminative learning methods, only the decision boundary is adjusted without forming a data generator in the entire feature space. In a discriminative learning scheme, the mapping function can be estimated using some criteria that are directly relevant to the ultimate classification and regression purpose, such as conditional maximum likelihood (CML) estimation (Jebara and Pentland, 1998) (a.k.a. maximum mutual information estimation (MMIE; Bahl et al., 1986; Woodland and Povey, 2002) in speech community), empirical risk minimization (ERM; Meir, 1995) and large margin estimation (LME; Scholkopf and Smola, 2002; Smola et al., 2000). Some representative discriminative models include logistic regression, regularization networks, support vector machines (SVM) and traditional neural networks. Particularly, based on the generalization bounds in statistical learning theory (Vapnik, 1998), large margin classifiers (Smola et al., 2000) have been attracting considerable attention in the field. However, there are still some limitations in the discriminative learning scheme. For example, it is not straightforward to deal with latent variables and exploit the underlying structure of data in discriminative models. Moreover, computational complexity is considerably higher in discriminative training since it requires simultaneous consideration of data from all classes. Particularly for ASR, many pure discriminative models, such as SVM, neural networks, logistic regression, have also been investigated for speech recognition but they fail to properly cope with the dynamic and variable-size nature of speech signals. Hence, no standalone discriminative model can yield comparable performance as generative models, i.e., HMMs, on any significant ASR task. The pure discriminative models are only used as a complementary component in HMM-based ASR systems, e.g., using neural networks as a front-end feature transformation module and SVM as a post-processing stage to combine scores.

More recently, an interesting topic has emerged in machine learning by combining both generative and discriminative learning schemes since they are regarded to be largely complementary, namely discriminative learning of generative models, such as (Altun et al., 2003; Jaakkola and Haussler, 1998; Jaakkola et al., 1999; Taskar et al., 2003). In a more general sense, discriminative training of generative models may include any alternative estimation methods for traditional generative models based on a different training criterion rather than MLE. This can be viewed as a general framework to learn generative models based on some discriminative criteria that are more consistent with the final pattern recognition and regression purpose. However, the existing generative learning algorithms, such as the EM algorithm, cannot be directly extended to optimize these alternative discriminative criteria. Over the past decade, considerable research efforts have been devoted to develop effective algorithms to be able to learn generative models discriminatively in the fields of ASR and machine learning. In this survey article, these research efforts will be comprehensively reviewed from the perspective of ASR, centering on discriminative learning of various types of HMMs. This work is also applicable to other generative models in different pattern classification areas and some of them may also be relevant to research work independently developed in machine learning.

In ASR, discriminative learning of HMMs has been extensively studied for several decades. Most research work in this category will be the main focus in this survey article. Since the middle of the 1980s, some IBM researchers (Nadas et al., 1988; Nadas, 1983) had started to theoretically study an alternative training method for HMMs, which was believed to be more pertinent to speech recognition task than the conventional MLE method. This method was initially posed from the perspective of information theory and was accordingly named as maximum mutual information estimation (MMIE; Bahl et al., 1986). It was later shown that under some minor conditions MMIE is in fact identical to the conditional maximum likelihood estimation (CMLE), a technique already known earlier. In Nadas et al. (1988), it was theoretically proved that the CMLE/MMIE method is superior to MLE when modeling assumptions are incorrect, which is obviously true for any practical application. Shortly after that, the IBM researchers had used a gradient descent method to implement CMLE/MMIE and shown that CMLE/MMIE based training produces less recognition errors than MLE in a small isolated word recognition task (Bahl et al., 1986; Brown, 1987). Following that, in Gopalakrishnan et al. (1991), they had continued to investigate a new optimization method based on growth transformation (GT), i.e., the extended Baum–Welch (EBW) algorithm, to implement CMLE/MMIE for DDHMMs in ASR. Later on, Normandin et al. (1994) had extended the EBW/GT method to CMLE/MMIE of Gaussian mixture CDHMMs and this method was used to successfully build a high-performance speaker-independent connected digit string recognition system, which was considered as an important ASR benchmark at that time. In the meantime, from early 1990s (Juang and Katagiri, 1992; Juang et al., 1997; Katagiri et al., 1998), some former Bell Labs researchers have also started to investigate a different discriminative training method for HMMs in ASR, which was named as minimum classification error (MCE) method. The MCE method aims to minimize an empirical error rate in training data which can be approximated by a smoothed and differentiable objective function. In Juang and Katagiri (1992), Juang et al. (1997), Katagiri et al. (1998), a generalized probabilistic descent (GPD) method has been proposed to optimize the MCE objective function, which is known as the MCE/GPD method. Similarly, the MCE/GPD method has also been shown to significantly outperform the MLE method in the connected digit recognition task (Chou et al., 1992; Chou et al., 1993). However, after that, discriminative training methods, including both CMLE/MMIE and MCE, failed to yield any significant improvement over the traditional MLE method except only on these relatively simple and small ASR tasks (Chou et al., 1993; Normandin et al., 1994). On the other hand, due to the effectiveness of the EM algorithm, the conventional MLE method has been successfully extended to the estimation of very large scale HMM models for large vocabulary ASR systems. The situation had not changed until very recently when Cambridge University researchers (Povey and Woodland, 2002; Woodland and Povey, 2002) first experimentally demonstrated that the MMIE/CMLE-based discriminative training can significantly improve well-trained MLE models even in the most challenging large vocabulary ASR tasks. In Woodland and Povey (2002), extensive experiments conducted in state-of-the-art ASR systems provide many useful insights to understand behaviors of discriminative training on large scale models and also give some methods to streamline practical implementation issues which are critically important for a successful implementation of discriminative training for large scale ASR systems. Since then, more and more promising results have been reported on discriminative training of HMMs for ASR. For example, some researchers have also reported significant gains to apply MCE-based

discriminative training to other large vocabulary ASR tasks, such as Jiang et al. (2005), Macherey et al. (2005), McDermott and Hazen (2004), McDermott et al. (2007). Nowadays, discriminative training techniques have been considered as the major driving force to bring down ASR errors from one level to another in almost all different kinds of applications and tasks.

In this survey article, we shall mainly review most relevant work in the literature regarding discriminative training of HMMs for speech recognition and highlight the most important theoretical points which are fundamental in discriminative learning of generative models and avoid technical and experimental details as much as possible. The remainder of this article is organized as follows. In Section 2, some important discriminative learning criteria for ASR are introduced, including MMIE/CMLE, MCE, minimum phone (word) error (MPE/MWE) and large margin estimation (LME), and a unifying view based on margin is presented to elucidate the relationship among these criteria. Next, in Section 3, several key optimization methods widely used in ASR are briefly reviewed. Then, as some recent advances, the so-called Approximation-optiMization (AM) method, is presented as a new general framework to solve discriminative training of HMMs in ASR. Under this general framework, several newly-proposed methods are also introduced, such as convex optimization and constrained line search and so on. In Section 5, some more practical issues regarding how to implement DT for large vocabulary continuous speech recognition (LVCSR) are discussed, particularly how to implement DT on word graphs. Finally, in Section 6, this article is concluded with some possible future research directions.

In another recent tutorial article on DT (He et al., 2008), the authors have reviewed recent research advances of DT in ASR mainly from aspect of discriminative criterion. It is shown that various types of DT criteria, including MMIE, MCE and MPE/MWE, can lead to the same form of objective functions, i.e., rational-function form. As a result, all of these DT criteria can be optimized using the same optimization method based on growth transformation. In this article, we treat both discriminative criteria and optimization methods in a more balanced way and present a different unifying view to survey all relevant DT works initially proposed from different contexts. As a technical survey, all technical methods and ideas are reviewed and discussed from high level without involving too much technical detail and experimental result, for which readers may refer to original papers based on a comprehensive reference list compiled at the end of this article.

2. Discriminative training criteria for ASR

Before we start to introduce discriminative training criteria, let's first clarify all necessary notations. Assume we view a sentence or word sequence S and its associated acoustic observation X (usually, a feature vector sequence) as a jointly distributed random variable pair (S,X). We denote the joint probability distribution as $p(S,X)$, which is normally represented by some pre-selected statistical models. In ASR, given any speech utterance X, an optimal speech recognizer chooses the sentence S^* which maximizes the posterior probability as output based on the plug-in MAP decision rule (Jiang et al., 1999) as follows:

$$S^* = \arg\max_S \; p(S|X) = \arg\max_S \; p(S)\cdot p(X|S) = \arg\max_S \; p(S)\cdot p(X|\lambda_S), \tag{1}$$

where λ_S denotes the composite HMM representing sentence or word sequence S. In this article, we are only interested in estimating HMM λ_S and assume language model used to calculate $p(S)$ is fixed. For convenience, we use Λ to denote the set of all HMM parameters that needs to be estimated in discriminative training (DT).

In supervised learning, given a set of training data, denoted as $\mathscr{D}$, consisting of many utterances as $\mathscr{D}=\{X_1,X_2,\cdots,X_T\}$, we usually know the true transcriptions for all utterances in $\mathscr{D}$, denoted as $\mathscr{L}=\{S_1,S_2,\cdots,S_T\}$. For notational convenience, we use the upper-case letter S_t to represent the true transcription of each utterance X_t, and use the lower-case letter s_t to denote a variable which may take all possible labels in a hypothesis space.

In the conventional maximum likelihood estimation (MLE), the HMM parameter set, Λ, is estimated by maximizing probability of training data $\mathscr{D}$ given their correct transcriptions $\mathscr{L}$. Under the *i.i.d.* assumption, the MLE criterion can be represented as

$$\widetilde{\Lambda}_{\mathrm{ML}} = \arg\max_\Lambda \; \Pr(\mathscr{D}|\mathscr{L},\Lambda) = \arg\max_\Lambda \; \prod_{t=1}^{T} p_\Lambda(X_t|S_t), \tag{2}$$

where $p_\Lambda(X_t|S_t)$ denotes probability of X_t given its correct transcript S_t, calculated with model Λ. For notational brevity, we will drop the subscript Λ hereafter.

As we know, the ML estimation of HMMs in Eq. (2) can be iteratively solved with the EM algorithm in Dempster et al. (1977), which leads to the well-known Baum-Welch (BW) algorithm (Baum et al., 1970).

2.1. Maximum mutual information estimation (MMIE)

In speech recognition, the first discriminative training criterion was derived from the perspective of information theory, which was named as maximum mutual information estimation (MMIE) accordingly (Bahl et al., 1986; Brown, 1987). The goal in MMIE is to maximize the mutual information between training data $\mathscr{D}$ and their corresponding labels $\mathscr{L}$ to establish the tightest possible relation (in a probabilistic sense) between training data and their corresponding models. However, as shown in Nadas et al. (1988), when the mutual information is calculated based on the sampling distribution of training data, MMIE is actually equivalent to another well-known training criterion, namely conditional maximum likelihood estimation (CMLE). In CMLE, the goal is to maximize the conditional probability of training labels $\mathscr{L}$ given the training data $\mathscr{D}$. Under the *i.i.d.* assumption, the CMLE criterion can be represented as

$$\widetilde{\Lambda}_{\mathrm{CML}} = \arg\max_{\Lambda} \Pr(\mathscr{L}|\mathscr{D}) = \arg\max_{\Lambda} \prod_{t=1}^{T} p(S_t|X_t) = \arg\max_{\Lambda} \prod_{t=1}^{T} \frac{p(S_t)\cdot p(X_t|S_t)}{\sum_{s_t} p(s_t)\cdot p(X_t|s_t)}, \tag{3}$$

where summation in denominator is conducted over all possible labels, s_t, for each X_t. Comparing with the MLE criterion in Eq. (2), it is clear that CMLE is more consistent with the MAP decision rule in Eq. (1) used in the final recognition, where each utterance is classified according to the same conditional probability. After taking logarithm on the above CMLE objective function, we derive the MMIE criterion widely used in speech recognition as

$$\widetilde{\Lambda}_{\mathrm{MMI}} = \arg\max_{\Lambda} \sum_{t=1}^{T} \log\left[\frac{p^{\kappa}(S_t)\cdot p^{\kappa}(X_t|S_t)}{\sum_{s_t} p^{\kappa}(s_t)\cdot p^{\kappa}(X_t|s_t)}\right], \tag{4}$$

where an exponential smoothing factor $\kappa(\kappa > 0)$ has been explicitly added to smooth the original MMIE objective function for effective optimization, see discussions in Section 3.4.

2.2. Minimum classification error (MCE)

The second discriminative training criterion in speech recognition, namely minimum classification error (MCE) estimation, has been developed to explicitly minimize the total error counts in training data (Juang et al., 1997; Katagiri et al., 1998). The key idea of MCE is to approximate the empirical classification errors in training data as a smoothed and differentiable objective function. In the classical MCE formulation, for each training data X_t in $\mathscr{D}$, the so-called mis-classification measure is first constructed as follows:

$$d_r(X_t, \Lambda) = -\log\left[p^{\kappa}(X_t|S_t)\cdot p^{\kappa}(S_t)\right] + \log\left[\sum_{s_t \neq S_t} p^{\kappa}(s_t)\cdot p^{\kappa}(X_t|s_t)\right], \tag{5}$$

where a similar smoothing factor $\kappa(\kappa > 0)$ is also introduced here, and the above summation is taken over all competing hypotheses $s_t(s_t \neq S_t)$ for X_t. In the above, log-sum is introduced as soft-max to determine the most competing hypothesis for X_t from all competing hypotheses. With κ properly set, we have $d_r(X_t, \Lambda) < 0$ if X_t is correctly recognized by Λ and $d_r(X_t, \Lambda) > 0$ otherwise. Traditionally, all possible competing hypotheses are given as an N-best list (Chou et al., 1992; Juang et al., 1997) so that the summation w.r.t. s_t is taken over the N-best list. Recently, it has been extended to represent all competing hypotheses as a word graph. In this case, s_t should be summed over all possible paths in the word graph.

Next, the above mis-classification measure is plugged into a sigmoid function to compute the so-called the smoothed error count for X_t as follows (Juang and Katagiri, 1992):

$$l_r(d_r(X_t,\Lambda)) = \frac{1}{1+e^{-d_r(X_t,\Lambda)}} \tag{6}$$

Finally, MCE aims to minimize the total smoothed error counts summed over the whole training set. Therefore, the MCE criterion can be represented as follows:

$$\widetilde{\Lambda}_{\mathrm{MCE}} = \arg\min_{\Lambda} \sum_{t=1}^{T} l_r(d_r(X_t,\Lambda)). \tag{7}$$

As shown in He et al. (2008), if we explicitly substitute the mis-classification measure in Eq. (5) into the sigmoid function in Eq. (6), after some manipulations, we can draw the MCE criterion into another equivalent form as follows:

$$\widetilde{\Lambda}_{\mathrm{MCE}} = \arg\max_{\Lambda} \sum_{t=1}^{T} \frac{p^{\kappa}(S_t)\cdot p^{\kappa}(X_t|S_t)}{\sum_{s_t} p^{\kappa}(s_t)\cdot p^{\kappa}(X_t|s_t)}. \tag{8}$$

2.3. Minimum phone (word) error (MPE/MWE)

As we know, the classification errors to be minimized in the above MCE formulation correspond to speech recognition errors measured in sentence level, i.e., string errors. However, in large vocabulary continuous speech recognition (LVCSR), recognition performance is normally measured in sub-string levels, e.g., word error rate (WER). Motivated by this, the work in Povey and Woodland (2002) has modified the MCE criterion to reflect sub-string errors in the following way:

$$\widetilde{\Lambda}_{\mathrm{MPE}} = \arg\max_{\Lambda} \sum_{t=1}^{T} \frac{\sum_{s_t} p^{\kappa}(s_t)\cdot p^{\kappa}(X_t|s_t)\cdot A(S_t,s_t)}{\sum_{s_t} p^{\kappa}(s_t)\cdot p^{\kappa}(X_t|s_t)}, \tag{9}$$

where $A(S_t,s_t)$ is called raw accuracy count, which is introduced to measure sub-string accuracy between two sentences S_t and s_t. For simplicity, $A(S_t,s_t)$ is pre-computed between any S_t and s_t and thus viewed as a constant coefficient in the above DT criterion. Unlike the MCE criterion in Eq. (8) which only considers the perfect string in numerators, a summation is conducted in numerators of the new objective function to consider all possible string labels, each of which is weighted by raw accuracy count. In this way, the new objective function includes the contribution from not only the perfect string but also many partially correct string labels. The raw accuracy count $A(S_t,s_t)$ can be calculated in several different sub-string levels. If it is computed in phone level, it represents phoneme accuracy between two sentences S_t and s_t. The resultant discriminative training criterion in Eq. (9) is called minimum phone error (MPE) estimation. Similarly, if $A(S_t,s_t)$ is calculated in word level, it represents word accuracy between two sentences S_t and s_t. The corresponding criterion in Eq. (9) is called minimum word error (MWE) estimation. As a special case, if $A(S_t,s_t)$ is calculated in sentence level, i.e., $A(S_t,s_t)=\delta(S_t-s_t)$, where $\delta(\cdot)$ stands for the Kronecker delta function, the criterion degenerates to the original MCE criterion in Eq. (8).

However, it is not trivial to calculate raw accuracy $A(S_t,s_t)$ in sub-string levels. Strictly speaking, it needs to use dynamic programing to compute the Levenshtein edit distance between S_t and s_t to account for substitution, deletion and insertion errors. Obviously, this kind of edit distances cannot be directly formulated in an objective function for optimization, except from simple N-best lists. Alternatively, as in Povey and Woodland (2002), $A(S_t,s_t)$ is normally calculated based on some simple heuristic measures which can be computed locally without dynamic programming; refer to Povey and Woodland (2002) for details.

2.4. Large margin estimation (LME)

More recently, a new discriminative training criterion has been proposed for speech recognition based on the principle of large margin classifiers, which is called large margin estimation (LME) criterion (Jiang, 2004; Jiang et al., 2006; Li and Jiang, 2007; Liu et al., 2005). LME aims to estimate HMM parameters based on the principle of maximizing minimum margin of training data towards better generalization capability and more

robustness in classifier design. In LME, we first define a separation margin for each training data, X_t, as follows:

$$d(X_t|\Lambda) = \log[p(S_t) \cdot p(X_t|S_t)] - \max_{s_t, s_t \neq S_t} \log[p(s_t) \cdot p(X_t|s_t)], \tag{10}$$

where *max* is taken over all competing hypotheses $s_t (s_t \neq S_t)$, which may be given as either an N-best list or a word graph. Similar to the mis-classification measure in MCE, we may use soft-max in the above margin definition, which results in a variant margin definition as follows:

$$d(X_t|\Lambda) = \log[p^\kappa(S_t) \cdot p^\kappa(X_t|S_t)] - \log\left[\sum_{s_t \neq S_t} p^\kappa(s_t) \cdot p^\kappa(X_t|s_t)\right]. \tag{11}$$

Obviously, $d(X_i|\Lambda) > 0$ if and only if X_i is correctly recognized by the model set Λ.

According to the statistical learning theory (Vapnik, 1998), the generalization error rate of a classifier in new test sets is theoretically bounded by a quantity related to margin. A large margin classifier usually yields lower error rate in new test sets and it shows more robustness and better generalization capability. Motivated by the large margin principle, even for those utterances in the training set with positive margin, we may still want to maximize their minimum margin to build an HMM-based large margin classifier.

The idea of large margin leads to estimating the HMM models Λ based on the criterion of maximizing the minimum margin of all training data as follows:

$$\widetilde{\Lambda}_{\text{LME}} = \arg\max_{\Lambda} \min_{X_t \in \mathscr{D}} d(X_t|\Lambda). \tag{12}$$

If we use the margin definition in Eq. (11), the LME criterion can be represented as follows:

$$\widetilde{\Lambda}_{\text{LME}} = \arg\max_{\Lambda} \min_{t=1\cdots T} \ln \frac{p^\kappa(S_t) \cdot p^\kappa(X_t|S_t)}{\sum_{s_t, s_t \neq S_t} p^\kappa(s_t) \cdot p^\kappa(X_t|s_t)}. \tag{13}$$

The above LME criterion is derived under the assumption that all training data is perfectly recognized by the current models. In case of training errors, the idea of soft-margin SVM can be applied to extend the LME criterion to consider training errors, such as the soft LME method in Jiang and Li (2007).

2.5. Discussions: DT criteria

After a quick investigation on their objective function forms, it is clear that all the above-mentioned discriminative training criteria are highly related to each other. In Macherey et al. (2005), Schluter (2000), several discriminative training (DT) criteria, including MMIE, MCE, MPE/MWE, are formulated in a general function form, which involves only different mapping and gain functions for different DT criteria. In He et al. (2008), it is shown that the objective functions derived from these discriminative criteria can all be converted into a general fraction form of two positive-valued functions, which provides a clear evidence that these different discriminative criteria can be optimized using the same optimization algorithm, such as the EBW method (to be discussed in Section 3.3).

In this section, we will propose yet another unifying view for all these discriminative training criteria, centering on the concept of margin. As in Eq. (11), assume that we adopt the definition of margin as the difference of log likelihood of the correct label versus that of the most competing hypothesis, which is selected over a hypothesis space based on *softmax* using log-sum. Given any training data X_t, its margin can be computed as follows:

$$d(X_t|\Lambda) = \log[p^\kappa(S_t) \cdot p^\kappa(X_t|S_t)] - \log\left[\sum_{s_t \in \mathscr{M}_t} p^\kappa(s_t) \cdot p^\kappa(X_t|s_t)\right], \tag{14}$$

where s_t is summed over a particular hypothesis space, denoted as $\mathscr{M}_t$.

Then, all discriminative objective functions can be represented as a general function of margins of all training samples in the training set, $\mathscr{D} = \{X_1, \cdots, X_T\}$. That is,

$$\mathbf{F}_{\text{DT}}(\Lambda) = \mathbf{f}\Big(d(X_1|\Lambda), d(X_2|\Lambda), \ldots, d(X_T|\Lambda)\Big). \tag{15}$$

For the MMIE or CMLE criterion in Section 2.1, the function $\mathbf{f}(\cdot)$ is a *sum* function. The objective function for MMIE or CMLE can be represented as

$$\mathbf{F}_{\mathrm{MMI}}(\Lambda) = \sum_{t=1}^{T} d(X_t|\Lambda), \tag{16}$$

where margin $d(X_t|\Lambda)$ is calculated as in Eq. (14) with s_t summed over all possible hypotheses (including the correct label S_t).

For the LME criterion in Section 2.4, the function $\mathbf{f}(\cdot)$ is a *min* function. The LME objective function can be written as

$$\mathbf{F}_{\mathrm{LME}}(\Lambda) = \min_{t=1,\cdots,T} d(X_t|\Lambda), \tag{17}$$

where margin $d(X_t|\Lambda)$ is still calculated as in Eq. (14) but with s_t summed only over all competing hypotheses (excluding the correct label S_t).

For the MCE criterion in Section 2.2, the function $\mathbf{f}(\cdot)$ is a *sum-exp* function. The MCE objective function can be expressed as

$$\mathbf{F}_{\mathrm{MCE}}(\Lambda) = \sum_{t=1}^{T} \exp\left[d(X_t|\Lambda)\right], \tag{18}$$

where margin $d(X_t|\Lambda)$ is calculated in the same way as MMIE, i.e., s_t is summed over all possible hypotheses (including the correct label).

For the MPE/MWE criterion in Section 2.3, the function $\mathbf{f}(\cdot)$ is still a *sum-exp* function but margin needs to be calculated in a slightly different way to incorporate all partially correct string labels for its positive term. Therefore, the MPE/MWE objective function can be represented as follows:

$$\mathbf{F}_{\mathrm{MPE}}(\Lambda) = \sum_{t=1}^{T} \exp\left[d'(X_t|\Lambda)\right], \tag{19}$$

with the variant margin $d'(X_t|\Lambda)$ calculated as

$$d'(X_t|\Lambda) = \log\left[\sum_{s_t \in \mathscr{M}_t} p^{\kappa}(s_t) \cdot p^{\kappa}(X_t|s_t) \cdot A(S_t, s_t)\right] - \log\left[\sum_{s_t \in \mathscr{M}_t} p^{\kappa}(s_t) \cdot p^{\kappa}(X_t|s_t)\right], \tag{20}$$

where $A(S_t, s_t)$ denotes the raw accuracy function as defined in Section 2.3, and the hypothesis space $\mathscr{M}_t$ consists of all possible string hypotheses, including correct transcript S_t.

3. Optimization methods for discriminative training in ASR

In the preceding section, we have briefly reviewed some popular discriminative training criteria for HMM-based speech recognition. In this part, we will summarize some important optimization methods proposed to optimize the objective functions constructed based on these criteria. As we have observed in recent work (McDermott et al., 2007; Schluter, 2000; Woodland and Povey, 2002), an effective optimization algorithm plays a crucial role in discriminative training for ASR. Especially in large vocabulary ASR, discriminative training methods need to deal with very large HMMs, which may result in optimization problems involving several millions of free variables. In practice, it is a huge challenge to solve this kind of large scale optimization problem efficiently and effectively and many important issues must be addressed appropriately, e.g., how to accelerate convergence speed and how to avoid bad shallow local optimum points and so on.

3.1. Gradient descent (GD)

In early discriminative training work for ASR (Bahl et al., 1986; Brown, 1987; Chou et al., 1992; Juang and Katagiri, 1992; Juang et al., 1997), it normally relies on a general gradient descent method to optimize the DT objective functions. The gradient descent method is simple and general and can be flexibly applied to any

differential objective functions. Given any objective function $\mathbf{F}(\Lambda)$, the general form of gradient descent search can be represented as the following iterative updating formula along the gradient direction:

$$\Lambda^{(n+1)} = \Lambda^{(n)} - \epsilon_n \cdot \nabla \mathbf{F}(\Lambda)|_{\Lambda=\Lambda^{(n)}}, \tag{21}$$

where ϵ_n is step size at n-th iteration, which should gradually decrease as iterations proceed. The above gradient descent search algorithm can be implemented in either batch or online mode. In batch mode, for each iteration, we normally accumulate gradient at $\Lambda^{(n)}$ over all training samples and update model parameters only once. The advantage of batch mode is that it is easy to parallelize optimization over multiple processors. In online mode, gradient is calculated for each single training sample and model parameters are immediately updated based on the gradient. The online gradient descent can automatically exploit data correlation, allowing learning to proceed quickly. The online method is also called probabilistic descent, which is a special case of stochastic approximation method. However, the online method is relatively slow to process a large amount of training data since it is hard to parallelize. In addition, the so-called 'semi-batch' mode is proposed as a compromise, where the model is updated every n training samples.

The major drawback of the gradient descent method lies in its very slow convergence speed since it only explores the first-order derivative, i.e., gradient, during the search process for the optimum. A uniform step size ϵ_n in Eq. (21) may not be appropriate for different model parameters. To ensure convergence for every parameter, an extremely small step size may have to be used in Eq. (21), which in turn leads to very slow convergence overall.

3.2. Quickprop, Rprop and Quasi–Newton methods

Obviously, we need to apply different step sizes to different model parameters to achieve better convergence speed. The second-order derivatives of the objective function $\mathbf{F}(\Lambda)$, i.e., the so-called Hessian matrix $H = \nabla^2 \mathbf{F}(\Lambda)$, provide the important information for properly setting different step sizes for different variables of the objective function.

In the traditional Newton's method, if the objective function can be approximated by a quadratic function and its Hessian matrix is positive definite, the optimum point, denoted as $\widetilde{\Lambda}$, can be reached from any starting point, $\Lambda^{(0)}$, in one single step along the gradient direction and the necessary step size can be calculated precisely based on the Hessian matrix:

$$\widetilde{\Lambda} = \Lambda^{(0)} - H^{-1} \cdot \nabla \mathbf{F}(\Lambda)|_{\Lambda=\Lambda^{(n)}}. \tag{22}$$

However, in practice there is no guarantee that the Hessian matrix is positive definite and also the size of the Hessian, i.e., the square of the number of model parameters, may prevent us from actually computing the Hessian.

In the literature, there exist many optimization methods which aim to approximate the Hessian matrix in various ways, such as Quasi–Newton, Quickprop, Rprop, BFGS and so on. The key idea behind these methods is to use a diagonal (or block diagonal) approximation of the Hessian matrix that can be efficiently updated over iterations. In this section, we briefly introduce Quickprop and Rprop since they both have been successfully applied to ASR.

The Quickprop method was initially proposed to train neural networks. In Quickprop, the Hessian is approximated by a diagonal matrix. The i-th diagonal element of Hessian at n-th iteration is approximately computed as finite difference of gradient as follows:

$$H_{ii} = \nabla^2_{ii} \mathbf{F}(\Lambda) = \frac{\partial^2 \mathbf{F}\left(\Lambda^{(n)}\right)}{\partial \lambda_i^2} \approx \frac{\frac{\partial \mathbf{F}\left(\Lambda^{(n)}\right)}{\partial \lambda_i} - \frac{\partial \mathbf{F}\left(\Lambda^{(n-1)}\right)}{\partial \lambda_i}}{\triangle \lambda_i^{(n-1)}}, \tag{23}$$

where $\triangle \lambda_i^{(n-1)}$ denotes the update step size of i-th parameter, λ_i, at previous iteration $n-1$. After substituting the diagonal Hessian approximation of Eq. (23) into the Newton's updating formula in Eq. (22), we derive the Quickprop updating formula for i-th parameter, λ_i, as

$$\lambda_i^{(n+1)} = \lambda_i^{(n)} - \triangle \lambda_i^{(n)} \cdot \nabla \mathbf{F}(\lambda_i)|_{\lambda_i=\lambda_i^{(n)}}, \tag{24}$$

where the step size for λ_i, i.e., $\triangle\lambda_i^{(n)}$, is calculated based on the approximated Hessian matrix as follows:

$$\triangle\lambda_i^{(n)} = \frac{\triangle\lambda_i^{(n-1)}}{\frac{\partial\mathbf{F}\left(\Lambda^{(n)}\right)}{\partial\lambda_i} - \frac{\partial\mathbf{F}\left(\Lambda^{(n-1)}\right)}{\partial\lambda_i}}. \tag{25}$$

Moreover, Quickprop also addresses the positive definiteness of the approximated Hessian by examining the sign of gradient w.r.t. each parameter for successive iterations, see McDermott et al. (2007) for details.

Similarly, Rprop also uses different step sizes to update different model parameters as in Eq. (24). In contrast, the updating step size in Rprop is determined based on only the sign of derivative, not the magnitude:

$$\triangle\lambda_i^{(n)} = \begin{cases} -\triangle_i^{(n)} & \text{if } \frac{\partial\mathbf{F}\left(\Lambda^{(n)}\right)}{\partial\lambda_i} > 0 \\ +\triangle_i^{(n)} & \text{if } \frac{\partial\mathbf{F}\left(\Lambda^{(n)}\right)}{\partial\lambda_i} < 0 \\ 0 & \text{otherwise,} \end{cases}$$

where the magnitude of step size, $\triangle_i^{(n)}$, is different for each parameter and evolves as follows:

$$\triangle_i^{(n)} = \begin{cases} \eta^+ \cdot \triangle_i^{(n-1)} & \text{if } \frac{\partial F\left(\Lambda^{(n-1)}\right)}{\partial\lambda_i}\frac{\partial F\left(\Lambda^{(n)}\right)}{\partial\lambda_i} > 0 \\ \eta^- \cdot \triangle_i^{(n-1)} & \text{if } \frac{\partial F\left(\Lambda^{(n-1)}\right)}{\partial\lambda_i}\frac{\partial F\left(\Lambda^{(n)}\right)}{\partial\lambda_i} < 0 \\ \triangle_i^{(n-1)} & \text{otherwise,} \end{cases}$$

with $0 < \eta^- < 1 < \eta^+$.

3.3. Extended Baum-Welch (EBW)

The most popular optimization method used for discriminative training of HMMs in ASR is the so-called extended Baum-Welch (EBW) method. The EBW algorithm was initially derived based on the concept of growth transformation in Baum and Eagon (1967), Baum and Sell (1968). As shown in Baum and Eagon (1967), Baum and Sell (1968), the so-called Baum–Eagon inequality can be used to construct a transformation which always increases the value of an arbitrary homogeneous polynomial function. As an alternative to the EM algorithm, this growth transformation can be used to estimate some discrete statistical models for MLE. In Gopalakrishnan et al. (1991), the Baum–Eagon inequality has been extended to any rational function, from which a growth transformation can be constructed for the MMIE objective function of discrete density HMMs (DDHMMs). It is also proved that the transformation can monotonically increase the MMIE objective function of DDHMMs under some conditions. This method is named as the EBW algorithm since its updating formula is reminiscent of the normal Baum-Welch algorithm for the MLE training. In Normandin et al. (1994), based on discrete approximation of the Gaussian distribution, the EBW method has been extended to Gaussian mixture continuous density HMMs (CDHMMs) without a rigid proof and the updating formula for MMIE training of Gaussian mixture CDHMMs has been derived accordingly. Since then, the derived EBW method has been widely used for discriminative training of HMMs in ASR (Kapadia, 1998; Povey, 2004; Valtchev, 1995). Until very recently, the work in Axelrod et al. (2007), Gunawardana and Byrne (2001), He et al. (2008) has finally given a mathematical proof that the derived EBW updating formula is guaranteed to strictly increase the MMIE objective function of Gaussian mixture CDHMMs under some conditions. Furthermore, in He et al. (2008), it has also been proved that the EBW formula can strictly increase other DT objective functions of CDHMMs, including MCE and MPE/MWE.

In the following, we briefly summarize the key results of the EBW method related to both DDHMMs and CDHMMs. Assume that a DT objective function, $\mathbf{F}$, involves some parameters of discrete statistical models, e.g., λ_{ij} with the sum-to-one constraint $\sum_j \lambda_{ij} = 1$ and $0 < \lambda_{ij} < 1$. The results in Gopalakrishnan et al. (1991), He et al. (2008) shows that the following re-estimation formula for λ_{ij}:

$$\lambda_{ij}^{(n+1)} = \frac{\lambda_{ij}^{(n)}\left(\frac{\partial\mathbf{F}}{\partial\lambda_{ij}}\Big|_{\lambda_{ij}=\lambda_{ij}^{(n)}} + D\right)}{\sum_k \lambda_{ik}^{(n)}\left(\frac{\partial\mathbf{F}}{\partial\lambda_{ik}}\Big|_{\lambda_{ik}=\lambda_{ik}^{(n)}} + D\right)}, \tag{26}$$

will converge to a local optimum of $\mathbf{F}$ for a sufficiently large value of constant D with the guarantee that $\mathbf{F}(\lambda_{ij}^{(n+1)}) \geqslant \mathbf{F}(\lambda_{ij}^{(n)})$.

On the other hand, if the DT objective function, $\mathbf{F}$, involves Gaussian distributions, e.g., $\mathcal{N}(\mu_{ik}, \Sigma_{ik})$ with mean vector μ_{ik} and covariance matrix Σ_{ik} for all i and k, the EBW re-estimation formula can also be derived as in He et al. (2008), Normandin et al. (1994). Moreover, a uniform EBW updating formula for all different DT criteria is presented in He et al. (2008). In this section, for simplicity, we take the MMIE objective function as an example to show its EBW updating formula. For other DT criteria, such as MCE, MPE/MWE, the updating formula has a similar form with some minor modifications. As in He et al. (2008), Normandin et al. (1994), the EBW updating formula for Gaussians based on MMIE can be derived as

$$\mu_{ik}^{(n+1)} = \frac{\mathcal{O}_{ik}^{\text{num}}(\mathbf{x}) - \mathcal{O}_{ik}^{\text{den}}(\mathbf{x}) + D\mu_{ik}^{(n)}}{\mathcal{O}_{ik}^{\text{num}}(1) - \mathcal{O}_{ik}^{\text{den}}(1) + D}, \tag{27}$$

$$\Sigma_{ik}^{(n+1)} = \frac{[\mathcal{O}_{ik}^{\text{num}}(\mathbf{x}\mathbf{x}^{\mathbf{t}}) - \mathcal{O}_{ik}^{\text{den}}(\mathbf{x}\mathbf{x}^{\mathbf{t}})] + D[\mu_{ik}^{(n)}\mu_{ik}^{(n)^t} + \Sigma_{ik}^{(n)}]}{\mathcal{O}_{ik}^{\text{num}}(1) - \mathcal{O}_{ik}^{\text{den}}(1) + D} - \mu_{ik}^{(n+1)}\mu_{ik}^{(n+1)^t}, \tag{28}$$

where $\mathcal{O}(1), \mathcal{O}(\mathbf{x})$ and $\mathcal{O}(\mathbf{x}\mathbf{x}^t)$ denotes occupancy statistics, data and squared data, collected for this Gaussian over time, and superscript *num* and *den* means statistics collected for numerators and denominators in the MMIE objective function respectively. As proved in Axelrod et al. (2007), Gunawardana and Byrne (2001), He et al. (2008), the updating formula is guaranteed to converge to a local optimum of the objective function if the constant D is sufficiently large. However, an important implementation issue in EBW is how to set the constant D. The results in Woodland and Povey (2002) suggest that different Gaussians should use different D values for better convergence. The works in Povey and Woodland (2002), Schluter (2000), He et al. (2008) give some good recipes to set Gaussian-specific D values for the EBW algorithm and these heuristic settings of D normally yield good and stable convergence behavior for various DT criteria.

As a remark, the work in Axelrod et al. (2007), He et al. (2008) has also shown that the EBW updating formula is comparable with an approximated quadratic Newtown search in terms of convergence behavior since an approximated Hessian has been implicitly used in EBW to determine step size for each parameter update, see Axelrod et al. (2007), He et al. (2008) for details. As a result, the EBW algorithm has achieved a huge success in discriminative training of HMMs for speech recognition. It has been widely used to optimize different DT criteria, including MMIE, MCE, MPE/MWE and so on.

3.4. Discussions: optimization methods

As we have mentioned, an effective optimization method plays the key role in discriminative training of HMMs for ASR, especially in large vocabulary ASR, where model size grows to be very large. To develop an effective optimization algorithm, the most important issues that need to be carefully addressed include: (i) how to accelerate convergence speed; (ii) how to avoid shallow local optimal points.

Towards good convergence in an iterative search algorithm, the key issue is how to set proper step sizes for updating different parameters. Instead of using a uniform step size for all parameters, it is important to use different step sizes for different parameters. The key information we need to set variant step sizes lies in second-order derivatives, i.e., the Hessian matrix. Because of this, the algorithms which explicitly or implicitly explore the second-order derivative information normally outperform, in terms of convergence speed, other methods using only the first-order derivatives. For example, the EBW method yields much better convergence performance than the simple gradient descent method. Moreover, in the EBW method, convergence speed can be further improved by setting different D values for different parameters.

As we know, all local search optimization methods only guarantee to converge to a local optimal point. If an objective function is highly complex and non-convex, its surface in space may be jagged and full of local optimal points all over the place. In this case, a local search algorithm may be quickly trapped into a bad shallow optimal point nearby the initial starting point. As the result, it becomes very difficult to observe any significant improvement in discriminative training. In ASR, this problem is largely resolved by introducing an exponential smoothing factor, $\kappa (0 < \kappa \ll 1)$, to smooth the originally derived DT objective functions, as shown in Section 2. The effect of κ (if κ is sufficiently small) is to flatten the objective functions and to get rid of most shallow local

optimal points. However, in the meantime, small κ values also make the smoothed objective functions largely deviating from the original discriminative criteria from which the objective functions are derived in the first place. It is clear that small κ values largely diminish discrepancy among all different DT objective functions discussed in Section 2.5 and make all of them have similar function surface for optimization. Another interesting observation is that the exponential smoothing using κ is actually related to the so-called deterministic annealing technique in Rose (1998), Rao and Rose (2001), where κ can be viewed as reciprocal of annealing temperature.

4. A new framework for discriminative training in ASR

In this section, we will introduce a new framework for discriminative training of HMMs in ASR based on some recent advances in the field (Jiang and Li, 2007; Jiang, 2007; Li and Jiang, 2005, 2007; Liu et al., 2005, 2008). The key idea of this framework is to cast the discriminative learning of HMMs as a locally constrained optimization problem and then iteratively approximate and optimize the DT objective functions within a close neighborhood. The new framework is fairly general and can be viewed as an extension of the conventional EM algorithm. It can be shown that some existing optimization methods, such as EBW, can be derived under this framework in a relatively simple way. More importantly, a number of new effective DT methods can be developed under this new framework.

4.1. Discriminative training as constrained optimization

As shown in Liu et al. (2007), Liu et al. (2008), during the optimization process of a general DT objective function, $\mathbf{F}(\Lambda)$, it is beneficial to impose a local constraint on model parameters Λ to ensure that they do not deviate too much from its current values, i.e., $\Lambda^{(n)}$. The local constraint can be quantitatively computed based on Kullback-Leibler divergence (KLD). Therefore, discriminative training of HMM parameters, Λ, can be formulated as the following iterative constrained maximization problem:

$$\Lambda^{(n+1)} = \arg\max_{\Lambda} \mathbf{F}(\Lambda) \tag{29}$$

$$\text{subject to} \quad \mathscr{D}\left(\Lambda||\Lambda^{(n)}\right) \leqslant \rho, \tag{30}$$

where $\mathscr{D}(\Lambda||\Lambda^{(n)})$ is the KLD between Λ and $\Lambda^{(n)}$, and $\rho > 0$ is a pre-set constant to control the search range. Apparently the constraint in Eq. (30) intuitively specifies a *trust region* for optimization in each iteration. As shown in Liu et al. (2008), for some models, such as Gaussians, the KLD-based constraint in Eq. (30) can be further relaxed as some quadratic contraints with the following form:

$$||\Lambda - \Lambda^{(n)}||_2 \leqslant \rho. \tag{31}$$

This quadratic constraint normally makes the constrained optimization much easier to solve and in some cases simple closed-form solutions may be derived.

As discussed in Liu et al. (2007), Liu et al. (2008), there are many reasons to justify the locality constraint imposed in optimization. First of all, the DT objective functions, $\mathbf{F}(\Lambda)$, in speech recognition, is highly complicated and nonlinear in nature, as discussed in Section 3.4, it is extremely difficult, if not impossible, to optimize them directly. Therefore, we normally make the following assumptions: (i) all competing hypotheses remain unchanged during optimization; (ii) all collected estimation statistics, such as state occupancies and Gaussian kernel occupancies, remain unchanged during optimization. The imposed locality constraint ensures these assumptions remain valid during optimization since the current model, $\Lambda^{(n)}$, has been used to generate all competing hypotheses and to accumulate statistics from training data prior to optimization. In addition, theoretical analysis of discriminative training algorithms in Afify et al. (2005), Afify et al. (2007) also supports using such a constraint in discriminative training.

4.2. A general Approximation-optiMization (AM) framework

Following Jiang and Li (2007), Jiang (2007), in this section, we introduce a general framework to solve the above-mentioned constrained maximization for discriminative training of HMMs in ASR. The key idea here is

that we first attempt to find a simpler auxiliary function to approximate the original DT function in a close proximity of current model parameters if the original objective function is too complicated to optimize directly. Then, the auxiliary function is optimized by using some efficient optimization algorithms. Because of the locality constraint in Eq. (30) or Eq. (31), we can apply a variety of approximation strategies to construct the auxiliary function with a simpler function form. Based on the proximity approximation, the optimal solution found for the approximate auxiliary functions is expected to improve the original objective function as well. Then, in next iteration, the original objective function can be similarly approximated in the close proximity of this new optimal solution based on the same approximation principle. This process repeats until convergence conditions are met for the original objective function. Analogous to the popular EM algorithm (Dempster et al., 1977; Neal and Hinton, 1998), each iteration consists of two separate steps: (i) Approximation step (**A-step**): the original objective function is approximated by an auxiliary function in a close proximity of current model parameters; (ii) optiMization step (**M-step**): the approximate auxiliary function is optimized under the locality constraints in either Eq. (30) or Eq. (31). Analogously, we call this method as the AM algorithm. It is clear that the AM algorithm is more general than the EM algorithm since the expectation (E-step) in EM can also be viewed as a proximity approximation method. More importantly, as shown below, the AM framework can also deal with some more complicated objective functions, such as those arising from discriminative training of many statistical models with hidden (or latent) variables.

4.2.1. Approximation step (A-step)

There are many different methods to construct auxiliary function to approximate an objective function in a close proximity. In this article, we only introduce an approximation strategy based on the Jensen's inequality. Readers can refer to Jiang and Li (2007), Jiang (2007) for other different approximation schemes.

As shown in Section 2.5, the DT objective functions, arising from discriminative training of statistical models, normally involve *log-sum* terms, which are typically difficult to deal with. Here, we consider a general strategy to use the well-known Jensen's inequality to approximate *log-sum*.

Assume that we have a finite number of positive-valued functions, i.e., $f_k(\Lambda)$, for $k = 1, \cdots, K$. For any fixed point Λ_0, we define the so-called posterior probability of kth function as

$$\xi_k(\Lambda_0) = \frac{f_k(\Lambda_0)}{\sum_{k=1}^{K} f_k(\Lambda_0)}.$$

Obviously, they satisfy the sum-to-one constraint that $\sum_{k=1}^{K} \xi_k(\Lambda_0) = 1$. According to the Jensen's inequality, for a function $\mathbf{F}(\Lambda)$ that is log-sum of all $f_k(\Lambda)$, we have the following inequality held for all Λ given Λ_0:

$$\begin{aligned}\mathbf{F}(\Lambda) &\equiv \ln\left[\sum_{k=1}^{K} f_k(\Lambda)\right] = \ln\left[\sum_{k=1}^{K} \xi_k(\Lambda_0)\frac{f_k(\Lambda)}{\xi_k(\Lambda_0)}\right] \geqslant \sum_{k=1}^{K} \xi_k(\Lambda_0)\ln\frac{f_k(\Lambda)}{\xi_k(\Lambda_0)} \equiv Q(\Lambda|\Lambda_0) \\ &= \sum_{k=1}^{K} \xi_k(\Lambda_0)\ln f_k(\Lambda) + H(\Lambda_0) \end{aligned} \tag{32}$$

where $H(\Lambda_0) = -\sum_{k=1}^{K} \xi_k(\Lambda_0)\ln\xi_k(\Lambda_0)$ denotes entropy calculated based on the posterior probabilities of $\xi_k(\Lambda_0)$. Furthermore, we can easily verify that:

$$\mathbf{F}(\Lambda)|_{\Lambda=\Lambda_0} = Q(\Lambda|\Lambda_0)|_{\Lambda=\Lambda_0} \tag{33}$$

and

$$\left.\frac{\partial \mathbf{F}(\Lambda)}{\partial \Lambda}\right|_{\Lambda=\Lambda_0} = \left.\frac{\partial Q(\Lambda|\Lambda_0)}{\partial \Lambda}\right|_{\Lambda=\Lambda_0}. \tag{34}$$

The above results suggest that $Q(\Lambda|\Lambda_0)$ can be viewed as a close proximity approximation of log-sum $\mathbf{F}(\Lambda)$ at Λ_0 with accuracy up to the first order. This approximation strategy is named as expectation-based approximation, i.e., **E-approx**, in Jiang and Li (2007), Jiang (2007).

As the first example, let us consider to use **E-approx** to approximate log likelihood function of HMMs in ASR, i.e., $\ln p(X_t, S_t|\Lambda)$ of X_t with transcription S_t:

$$\ln p(X_t, S_t|\Lambda) = \ln \sum_{\mathbf{l}} p(X_t, S_t, \mathbf{l}|\Lambda). \tag{35}$$

where $\mathbf{l}$ denotes all missing data in HMM, such as hidden state sequences and unknown mixture labels.

Following Eq. (32), $\ln p(X_t, S_t|\Lambda)$ can be approximated by the following auxiliary function at a close proximity of $\Lambda^{(n)}$:

$$\begin{aligned}\mathcal{Q}(\Lambda|\Lambda^{(n)}) &= \mathbb{E}_{\mathbf{l}}\left[\ln p(X_t, S_t, \mathbf{l}|\Lambda)|X_t, S_t, \Lambda^{(n)}\right] + H_{\mathbf{l}}(\Lambda^{(n)}) \\ &= \sum_{\mathbf{l}} \ln p(X_t, S_t, \mathbf{l}|\Lambda) \cdot \Pr(\mathbf{l}|X_t, S_t, \Lambda^{(n)}) + H_{\mathbf{l}}(\Lambda^{(n)}).\end{aligned} \tag{36}$$

Obviously, if we use this approximation for MLE of HMMs, it results in the well-known Baum–Welch (or EM) algorithm. Since the conditions in Eqs. (32) and (33) hold, it is straightforward to prove that any increase in $\mathcal{Q}(\Lambda|\Lambda^{(n)})$ also means an even bigger increase in the original log likelihood function.

In the following, we will consider to use **E-approx** to approximate various DT objective functions of HMMs in ASR. As discussed in Section 2.5, all DT objective functions can be viewed as a function of margins, $d(X_t|\Lambda)$, of all training data. Therefore, we first consider to use **E-approx** to approximate the margin, which is defined as difference of log likelihood of correct model versus that of incorrect competing models. Following the same idea in Eq. (36), log likelihood of correct model, i.e., $p(X_t, S_t|\Lambda)$, can be approximated by an auxiliary function, denoted as $\mathcal{Q}_t^+(\Lambda|\Lambda^{(n)})$:

$$\ln p(X_t, S_t|\Lambda) \approx \mathcal{Q}_t^+\left(\Lambda|\Lambda^{(n)}\right). \tag{37}$$

Similarly, we can also use **E-approx** to approximate the log likelihood function of incorrect competing models as follows:

$$\begin{aligned}\ln \sum_{s_t \in \mathcal{M}_t} p(X_t, s_t|\Lambda) &= \ln \sum_{s_t \in \mathcal{M}_t} \sum_{\mathbf{l}} p(X_t, s_t, \mathbf{l}|\Lambda) \approx \mathcal{Q}_t^-\left(\Lambda|\Lambda^{(n)}\right) \\ &\equiv \sum_{s_t \in \mathcal{M}_t} \sum_{\mathbf{l}} \ln p(X, s_t, \mathbf{l}|\Lambda) \cdot \Pr\left(\mathbf{l}|X_t, s_t, \Lambda^{(n)}\right) + H_{s_t\mathbf{l}}\left(\Lambda^{(n)}\right),\end{aligned} \tag{38}$$

where s_t is summed over a hypothesis space $\mathcal{M}_t$ of competing hypotheses, which may be represented as either N-best list or word graph, and $\Pr(\mathbf{l}|X_t, s_t, \Lambda^{(n)})$ denotes the posterior probability of missing data $\mathbf{l}$ based on one competing hypothesis s_t.

It can be easily shown that both $\mathcal{Q}_t^+(\Lambda|\Lambda^{(n)})$ and $\mathcal{Q}_t^-(\Lambda|\Lambda^{(n)})$ are concave functions for HMMs Λ. If language model scores are assumed to be constant, the decision margin of HMMs can be similarly approximated as difference of two concave functions as follows:

$$d(X_t|\Lambda) \approx \mathcal{Q}_t^+\left(\Lambda|\Lambda^{(n)}\right) - \mathcal{Q}_t^-\left(\Lambda|\Lambda^{(n)}\right), \tag{39}$$

for any training sample X_t.

4.2.1.1. LME. Therefore, based on **E-approx** of margin in Eq. (39), the LME objective function in Section 2.5 can be approximated as follows:

$$\mathbf{F}_{\mathrm{LME}}(\Lambda) = \min_t\ d(X_t|\Lambda) \approx \mathcal{Q}_{\mathrm{LME}}\left(\Lambda|\Lambda^{(n)}\right) = \min_t\ \left[\mathcal{Q}_t^+\left(\Lambda|\Lambda^{(n)}\right) - \mathcal{Q}_t^-\left(\Lambda|\Lambda^{(n)}\right)\right], \tag{40}$$

where $\mathcal{Q}_{\mathrm{LME}}(\Lambda|\Lambda^{(n)})$ denotes the auxiliary function used to approximate the original LME objective function based on **E-approx**.

Obviously, *min* in Eq. (40) can not be directly optimized. In Jiang et al. (2006), Li et al. (2005), it is approximated by soft-max based on *log-sum*. In Li and Jiang (2006), Li and Jiang (2007), a new variable is introduced to convert *min* into some equivalent constraints in optimization.

4.2.1.2. MMIE. As in Section 2.5, the MMIE objective function can be represented as summation of margins over all training data. Therefore, the MMIE objective function can be approximated as follows:

$$\mathbf{F}_{\mathrm{MMI}}(\Lambda) = \sum_t d(X_t|\Lambda) \approx \mathcal{Q}_{\mathrm{MMI}}\left(\Lambda|\Lambda^{(n)}\right) = \sum_{t=1}^{T} \mathcal{Q}_t^+\left(\Lambda|\Lambda^{(n)}\right) - \sum_{t=1}^{T} \mathcal{Q}_t^-\left(\Lambda|\Lambda^{(n)}\right), \tag{41}$$

where $\mathcal{Q}_{\mathrm{MMI}}(\Lambda|\Lambda^{(n)})$ denotes the auxiliary function of the MMIE objective function based on **E-approx**.

4.2.1.3. MCE/MPE/MWE. As shown in Section 2.5, the MCE/MPE/MWE objective functions can be represented as an exponential summation of margins over all training data, where margin is defined in a slightly different way with different weighting schemes over all possible hypotheses. Similarly, these objective functions can also be approximated under the AM framework. Here, as an example, we only show how to use **E-approx** to approximate the MCE objective function. Similarly, we can apply the same procedure to MPE/MWE as well.

As in Section 2.5, the MCE objective function can be represented as an exponential sum of margins as

$$\mathbf{F}_{\mathrm{MCE}}(\Lambda) = \exp\left\{\ln \sum_{t=1}^{T} \exp\left[d(X_t|\Lambda)\right]\right\}. \tag{42}$$

In order to construct a simple auxiliary function for MCE, we first need to use **E-approx** to approximate *log-sum* appearing in Eq. (42), and then use **E-approx** to approximate margin $d(X_t|\Lambda)$ as in Eq. (39).

First of all, we define an MCE weighting factor for each training sample X_t based on a given model $\Lambda^{(n)}$ as

$$\phi_t\left(\Lambda^{(n)}\right) = \frac{p\left(X_t, S_t|\Lambda^{(n)}\right)\prod_{u\neq t}\sum_{s_u} p\left(X_u, s_u|\Lambda^{(n)}\right)}{\sum_t\left[p\left(X_t, S_t|\Lambda^{(n)}\right)\prod_{u\neq t}\sum_{s_u} p\left(X_u, s_u|\Lambda^{(n)}\right)\right]}. \tag{43}$$

Therefore, the original MCE objective function in Eq. (42), $\mathbf{F}_{\mathrm{MCE}}(\Lambda)$, can be approximated by the following auxiliary function:

$$\mathcal{Q}_{\mathrm{MCE}}\left(\Lambda|\Lambda^{(n)}\right) = \sum_{t=1}^{T} \phi_t\left(\Lambda^{(n)}\right)\cdot\left[\mathcal{Q}_t^{+}\left(\Lambda|\Lambda^{(n)}\right) - \mathcal{Q}_t^{-}\left(\Lambda|\Lambda^{(n)}\right)\right]. \tag{44}$$

Comparing Eq. (44) with Eq. (41), we can see that the MCE auxiliary function, $\mathcal{Q}_{\mathrm{MCE}}(\Lambda|\Lambda^{(n)})$, has a similar function form as the MMIE auxiliary function, $\mathcal{Q}_{\mathrm{MMI}}(\Lambda|\Lambda^{(n)})$. The only difference is that all training samples contribute equally in $\mathcal{Q}_{\mathrm{MMI}}$ while they are weighted by $\phi_t(\Lambda^{(n)})$ in $\mathcal{Q}_{\mathrm{MCE}}$.

4.2.2. Optimization step (M-step)

Under the proximity constraint in Eq. (30) or Eq. (31), the above **E-approx** auxiliary functions serve as good approximation of the original DT objective function within the locality constraint. If the basic statistical models belong to the exponential family, it is clear that the auxiliary functions have a much simpler function form, whose optimal point can be found in a relatively simple way. Due to the proximity constraint, we expect that the found optimal solution also improves the original DT objective function since the auxiliary functions approaches the original DT functions with sufficient accuracy under the proximity constraint in Eq. (30) or Eq. (31). In some situations, if we can formulate the approximated auxiliary function as a strict lower bound of the original DT functions (as in Section 4.3), then it is guaranteed that the found optimal point of the auxiliary function will strictly increase the original DT objective function.

However, unlike the EM algorithm, we still encounter serious difficulties in optimizing these auxiliary functions since all of them involve difference of two concave functions, i.e., $\mathcal{Q}^{+} - \mathcal{Q}^{-}$. As a result, these auxiliary functions are neither convex nor concave, which makes this optimization step still a huge challenge in most cases. Of course, a variety of methods can be applied to solve this non-convex optimization problem. In the following, we will introduce several techniques which have been successfully applied to solve this non-convex optimization problem in the **M-step**.

4.3. Deriving EBW under the AM framework

Since most widely-used statistical models belong to the exponential family, it is easy to show that the auxiliary functions, $\mathcal{Q}_{\mathrm{DT}}$, are in fact multivariate polynomial functions, but they are normally neither convex nor concave due to the involved negative terms. An easy method to make them concave is to add one extra negative quadratic term as

$$\mathbf{F}_{\mathrm{DT}}(\Lambda) \approx \mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0) - D \cdot ||\Lambda - \Lambda_0||_2 \tag{45}$$

where D is a positive constant. We denote this new auxiliary function as $\mathcal{E}_{\mathrm{DT}}(\Lambda|\Lambda_0)$, i.e.,

$$\mathcal{E}_{\mathrm{DT}}(\Lambda|\Lambda_0) \equiv \mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0) - D \cdot ||\Lambda - \Lambda_0||_2.$$

It is straightforward to show that this new auxiliary function still satisfies the tangential constraints with $\mathbf{F}_{\mathrm{DT}}$ at Λ_0, i.e., Eqs. (33) and (34). Moreover, if D is large enough, $\mathcal{E}_{\mathrm{DT}}(\Lambda|\Lambda_0)$ serves as a lower bound of $\mathbf{F}_{\mathrm{DT}}(\Lambda)$ for any Λ as

$$\mathcal{E}_{\mathrm{DT}}(\Lambda|\Lambda_0) \leqslant \mathbf{F}_{\mathrm{DT}}(\Lambda) \text{ if } D \text{ is sufficiently large.} \tag{46}$$

More importantly, if D is large enough, the negative quadratic term in $\mathcal{E}_{\mathrm{DT}}(\Lambda|\Lambda_0)$ compensates for all positive elements in $\mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0)$, making $\mathcal{E}_{\mathrm{DT}}(\Lambda|\Lambda_0)$ a strict concave function. If $\mathcal{E}_{\mathrm{DT}}(\Lambda|\Lambda_0)$ is concave, its global maximal point can be found by making its derivatives vanish as follows:

$$\frac{\partial \mathcal{E}_{\mathrm{DT}}(\Lambda|\Lambda_0)}{\partial \Lambda} = \frac{\partial \mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0)}{\partial \Lambda} - 2D \cdot (\Lambda - \Lambda_0) = 0. \tag{47}$$

For HMMs or other exponential family models, the equation in Eq. (47) can be easily solved and the global maximum of $\mathcal{E}_{\mathrm{DT}}$, denoted as Λ^*, can be derived with a simple closed-form solution. Once Λ^* is derived, it satisfies $\mathcal{E}_{\mathrm{DT}}(\Lambda^*|\Lambda_0) \geqslant \mathcal{E}_{\mathrm{DT}}(\Lambda_0|\Lambda_0)$ since Λ^* is globally maximal. Furthermore, we have $\mathbf{F}_{\mathrm{DT}}(\Lambda^*) > \mathcal{E}_{\mathrm{DT}}(\Lambda^*|\Lambda_0)$ since $\mathcal{E}_{\mathrm{DT}}$ is a strict lower bound of $\mathbf{F}_{\mathrm{DT}}$, and $\mathbf{F}_{\mathrm{DT}}(\Lambda_0) = \mathcal{E}_{\mathrm{DT}}(\Lambda_0|\Lambda_0)$ based on Eq. (33). Finally, we have $\mathbf{F}_{\mathrm{DT}}(\Lambda^*) \geqslant \mathbf{F}_{\mathrm{DT}}(\Lambda_0)$. In other words, it is guaranteed that Λ^* always increases the original objective function provided the constant D is sufficiently large.

It is interesting that solving Eq. (47) leads to the well-known EBW updating formula in Section 3.3. For example, if we substitute the MMIE auxiliary function, $\mathcal{Q}_{\mathrm{MMI}}$, of Gaussian mixture CDHMMs into Eq. (47), we will derive the same updating formula for mean vectors and covariance matrices as in Eqs. (27) and (28). Obviously, the discussions in this section serve as another strict mathematical proof for the convergence of EBW, i.e., the EBW update formula leads to improve the original DT objective function as long as the constant D is large enough.

As a remark, in Afify (2005), the EBW estimation formula for MMIE has also been derived based on a similar idea of optimizing a lower bound of the MMIE objective function. The major difference is that the lower bound in Afify (2005) is derived according to the reverse Jensen's inequality (Jebara and Pentland, 2000).

4.4. Convex relaxations: LP, SDP and SOCP

As opposed to the above simple method in Section 4.3 that compensates non-convex auxiliary functions by adding a large negative quadratic term, a variety of convex relaxation methods (Li, 2005; Li and Jiang, 2007; Pan and Jiang, 2008; Pan, 2008; Yin and Jiang, 2007; Yin, 2007) can be used to convert the non-convex optimization problem in **M-step** into a standard convex optimization problem, such as linear programming (LP), Quadratic programming (QP), second-order cone programming (SOCP) and semi-definite programming (SDP), so that some standard convex optimization algorithms can be applied to optimize the relaxed auxiliary functions under the proximity constraint in Eqs. (30) or (31). As we know, any local optimal point is always globally optimal in a convex optimization problem. As a result, a convex optimization problem can be efficiently solved even in a very high-dimensionality space since it never suffers from the local optimum problem. Therefore, the advantage of using convex optimization in **M-step** is that optimization can be efficiently and reliably solved even for very large scale models.

As in Pan and Jiang (2008), Pan (2008), the DT objective functions of various discrete statistical models based on multinomial distribution, such as multinomial mixture model, Markov chain model, discrete density HMMs and so on, can be approximated with **E-approx** as linear auxiliary functions. Then, in **M-step**, optimization of these linear auxiliary functions can be converted into a standard linear programming problem if the sum-to-one constaint is relaxed.

As shown in Jiang and Li (2007), Li (2005), Li and Jiang (2007), the DT objective function of many Gaussian-derived statistical models, such Gaussian mixture model (GMM), Gaussian mixture CDHMMs and so on,

can be approximated with **E-approx** as quadratic auxiliary functions. As in Li (2005), Li and Jiang (2007), Yin and Jiang (2007), maximization of these non-convex quadratic functions can be represented as a matrix form. If the self-constrained matrix variables can be relaxed as positive semi-definite matrices, the original non-convex maximization problem in **M-step** can be converted into a semi-definite program (SDP) problem. In Yin and Jiang (2007), Yin (2007), a different approach is taken to convert maximization of indefinite quadratic form to convex optimization, where the indefinite Hessian matrix is decomposed based on eigenvectors with positive and negative eigenvalues. All quadratic terms related to negative eigenvalues are replace by a linear term along with some convex constraints. In this way, the original problem to maximize indefinite quadratic form can be relaxed into another convex optimization problem, namely second order cone programming (SOCP).

4.5. Constrained line search (CLS)

In Liu et al. (2007), Liu et al. (2008), instead of using convex optimization, a constrained line search (CLS) method is proposed to solve the non-convex optimization problem in **M-step**. In CLS, the original objective function, $\mathbf{F}_{\mathrm{DT}}(\Lambda)$, is first approximated by a quadratic function, $\mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0)$, based on **E-approx**. Then, for any model parameter λ, the critical point of $\mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0)$, denoted as $\hat{\lambda}$, can be easily derived with a closed-form solution by equating its derivative to zero, i.e., $\frac{\partial}{\partial\lambda}\mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0) = 0$. However, the found critical point $\hat{\lambda}$ may be: (i) a saddle point if $\mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0)$ is indefinite w.r.t. λ; (ii) a local minimum if $\mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0)$ is positive definite w.r.t. λ; (iii) a local maximum if $\mathcal{Q}_{\mathrm{DT}}(\Lambda|\Lambda_0)$ is negative definite w.r.t. λ. Even though the critical point is a local maximum, it may be located too far away from the current model so that the locality constraint in Eq. (31) is not satisfied. In Liu et al. (2007), Liu et al. (2008), a line search method is proposed to maximize the objective function $\mathbf{F}_{\mathrm{DT}}(\Lambda)$ along the direction of line segment joining the current model and the found critical point under the locality constraint in Eq. (31). As shown in Liu et al. (2007), Liu et al. (2008), if the quadratic locality constraint in Eq. (31) is used, the line search can be efficiently solved with a closed-form solution. More specifically, the model parameter λ is updated along direction $\boldsymbol{d}$ and step size ϵ as follows:

$$\lambda^{(n+1)} = \lambda^{(n)} + \epsilon \cdot \boldsymbol{d}, \tag{48}$$

where the search direction $\boldsymbol{d}$ is determined as line segment joining $\lambda^{(n)}$ and $\hat{\lambda}$, i.e., $\hat{\lambda} - \lambda^{(n)}$, or gradient $\nabla\mathbf{F}(\lambda^{(n)})$ if the critical point is a saddle point, and the optimal step size ϵ is determined based on the quadratic constraint in Eq. (31).

5. Discriminative training for LVCSR

As mentioned before, discriminative training (DT) has been successfully applied to not only small vocabulary ASR tasks but also very large scale ASR tasks (Woodland and Povey, 2002). In this section, we will briefly discuss some practical issues to implement DT for large vocabulary continuous speech recognition (LVCSR).

When we implement the above-mentioned discriminative training methods for LVCSR, the most important issue is how to represent the overall hypothesis space, where a competing string s_t needs to sum over in a DT objective function. In some early work, the hypothesis space is given as a list, i.e., the so-called N-best list, which includes the top N most competing string hypotheses for each training sample X_t. In this case, we have no technical difficulty to implement discriminative training since we just need to sum s_t over all competing hypothesis strings in this finite list. However, it has been found that N-best list is not a good way to represent the competing hypothesis space, especially in LVCSR.

In most recent DT work, word graphs (a.k.a. word lattices) have been widely adopted to represent the competing hypothesis space for DT. A word graph is represented as a directed, acyclic, weighted graph. Its nodes represent discrete points in time. Each arc, denoted as a, is labeled with three variables, i.e., $a = [w]_s^e$, where w is the hypothesized word attached to this arc, and s and e denote the starting and ending time instances of the arc, respectively. Also, each arc is associated with a weight, which is actually acoustic score to generate acoustic feature vectors from time s to e based on HMM of word w. In a word graph, there are two special nodes: one is called START node which corresponds to the beginning of the utterance and one END node for the end

of the utterance. Any path from START node to END node is called a complete path which represents a sentence (a sequence of words) hypotheses for the underlying utterance. Obviously, word graph is a very compact method to represent hypothesis strings since even a relatively small graph may include a large number of string hypotheses. When a word graph is used to represent the hypothesis space for DT objective functions, competing string s_t needs to sum over all complete paths in the word graph. If we directly apply **E-approx** to construct auxiliary functions for word graphs, calculation of $\mathcal{Q}^-(\cdot)$ requires summation of posterior probabilities for all complete paths in the word graph, which obviously is computationally prohibitive. However, it is easy to show that the summation can be re-arranged over all arcs in the graph, instead of all complete paths. We first run the forward–backward algorithm in Wessel et al. (2001) to obtain the posterior probability for every arc. Then we run the forward–backward algorithm locally for every arc to collect statistics from this arc. At last, we sum up statistics from all arcs, weighted by the pre-calculated posterior probability of the arc, to compute the final overall statistics from the entire word graph. In this way, the auxiliary function $\mathcal{Q}_{\mathrm{DT}}(\cdot)$ can be computed fairly efficiently even for very large graphs.

Finally, all word graphs used for DT are generated from a Viterbi beam search. Some experimental results suggest that a rather weak language model (such as unigram or bigram) should be used in order to generate more dense graphs to cover more acoustically competing hypotheses for DT.

6. Conclusions and future work

Discriminative training has achieved a huge success in ASR during the past decades. Specially in recent years, discriminative training methods have steadily driven down speech recognition error rates across a variety of tasks. In this paper, we have reviewed these discriminative training techniques for HMMs in context of acoustic modeling in ASR. We believe discriminative training will continue to be a very active research area in ASR and expect that discriminative training methods will be extended to every corner of recognizer design, such as discriminative training of language models, discriminative learning of model structure not just parameters, discriminative feature extraction in the front-end. At last, we also anticipate that the successful story of discriminative training in ASR will soon inspire more applications of discriminative learning techniques in other domains, such as statistical machine translation, text categorization, computer vision, biometrics, informatics and much more.

References

Afify, M., Li, X.W., Jiang, H., 2005. Statistical performance analysis of MCE/GPD learning in gaussian classifiers and hidden Markov models. In: Proceedings of ICASSP-05, Philadelphia, Pennsylvania.

Afify, M., 2005. Extended Baum–Welch reestimation of gaussian mixture models based on reverse jensen inequality. In: Proceedings of Interspeech 2005, Lisboa.

Afify, M., Li, X.-W., Jiang, H., 2007. Statistical analysis of minimum classification error learning for gaussian and hidden Markov model classifiers. IEEE Transactions on Audio, Speech and Language Processing 15 (8), 2405–2417.

Altun, Y., Tsochantaridis, I., Hofmann, T., 2003. Hidden Markov support vector machines. In: Proceedings of the 20th International Conference on Machine Learning (ICML-2003), Washington, DC.

Axelrod, S., Goel, V., Gopinath, R., Olsen, P., Visweswariah, K., 2007. Discriminative estimation of subspace constrained Gaussian mixture models for speech recognition. IEEE Transactions on Audio, Speech and Language Processing 15 (1), 172–189.

Bahl, L.R., Brown, P.F., De Souza, P.V., Mercer, R.L., 1986. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In: Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'86), Tokyo, Japan, pp. 49–52.

Baum, L.E., Eagon, J.A., 1967. An inequality with applications to statistical prediction for functions of Markov processes and to a model of ecology. Bulletin of the American Mathematical Society 73, 360–363.

Baum, L.E., Sell, G., 1968. Growth transformation for functions on manifolds. Pacific Journal of Mathematics 27 (2), 211–227.

Baum, L.E., Petrie, T., Soules, G., Weiss, N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Annals of Mathematical Statistics 41 (1), 164–171.

Brown, L.D., 1986. Fundamentals of Statistical Exponential Families, with Applications in Statistical Decision Theory. Institute of Mathematical Statistics, Hayward, California.

Brown, P., 1987. The acoustic modeling problem in automatic speech recognition, Ph.D. Dissertation, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Chou, W., Juang, B.-H., Lee, C.-H., 1992. Segmental GPD training of HMM based speech recognition. In: Proceedings of IEEE ICASSP92, vol. 1. pp. 473–476.

Chou, W., Juang, B.-H., Lee, C.-H., 1993. Minimum error rate training based on *N*-best string models. In: Proceedings of IEEE ICASSP93, vol. 2. pp. 652–655.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). Journal of the Royal Statistical Society B 39, 1–38.

Gopalakrishnan, P.S., Kanevsky, D., Nadas, A., Nahamoo, D., 1991. An inequality for rational functions with applications to some statistical estimation problems. IEEE Transactions on Information Theory 37 (1), 107–113.

Gunawardana, A., Byrne, W., 2001. Discriminative speaker adaptation with conditional maximum likelihood linear regression. In: Proceedings of Eurospeech 2001.

He, X., Deng, L., Chou, W., 2008. Discriminative learning in sequential pattern recognition: a unifying review for optimization-based speech recognition. IEEE Signal Processing Magazine, 14–36.

Jaakkola, T., Haussler, D., 1998. Exploiting generative models in discriminative classifiers. In: Proceedings of Advances in Neural Information Processing Systems, vol. 11.

Jaakkola, T., Meila, M., Jebara, T., 1999. Maximum entropy discrimination. In: Proceedings of Advances in Neural Information Processing Systems, vol. 12.

Jebara, T., Pentland, A., 1998. Maximum conditional likelihood via bound maximization and the CEM algorithm. In: Proceedings of Advances in Neural Information Processing Systems, vol. 11.

Jebara, T., Pentland, A., 2000. On reversing Jensen's inequality. In: Proceedings of NIPS'2000.

Jebara, T., 2002. Discriminative, Generative and Imitative Learning, Ph.D. Thesis, MIT.

Jiang, H., Hirose, K., Huo, Q., 1999. Robust speech recognition based on Bayesian prediction approach. IEEE Transactions on Speech and Audio Processing 7 (4), 426–440.

Jiang, H., 2004. Discriminative training for large margin HMMs, Technical Report CS-2004-01, Department of Computer Science and Engineering, York University.

Jiang, H., Soong, F., Lee, C.-H., 2005. A dynamic in-search data selection method with its applications to acoustic modeling and utterance verification. IEEE Transactions on Speech and Audio Processing 13 (5), 945–955.

Jiang, H., Li, X., Liu, C.-J., 2006. Large Margin Hidden Markov Models for Speech Recognition. IEEE Transactions on Audio, Speech and Language Processing 14 (5), 1584–1595.

Jiang, H., Li, X., 2007. Incorporating training errors for large margin HMMs under semi-definite programming framework. In: Proceedings of 2007 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2007), Hawaii, USA.

Jiang, H., Li, X., 2007. A general approximation–optimization approach to large margin estimation of HMMs. In: V. Kodic (Ed.), Speech Recognition and Synthesis.

Jiang, H., 2007. A general formulation for discriminative learning of graphical models, Technical Report, Department of Computer Science and Engineering, York University, Toronto, Canada.

Jordan, M.I., 2004. Graphical models. Statistical Science 19 (Special Issue on Bayesian Statistics), 140–155.

Juang, B.-H., Katagiri, S., 1992. Discriminative learning for minimum error classification. IEEE Transactions on Signal Processing 40, 3043–3054.

Juang, B.-H., Chou, W., Lee, C.-H., 1997. Minimum classification error rate methods for speech recognition. IEEE Transactions on Speech and Audio Processing 5 (3), 257–265.

Kapadia, S., 1998. Discriminative training of hidden Markov models, Ph.D. Dissertation, Engineering Department, Cambridge University, UK.

Katagiri, S., Juang, B.-H., Lee, C.-H., 1998. Pattern recognition using a generalized probabilistic descent method. Proceedings of the IEEE 86 (11), 2345–2373.

Li, X., Jiang, H., Liu, C.-J., 2005. Large margin HMMs for speech recognition. In: Proceedings of 2005 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2005), Philadelphia, Pennsylvania, pp.V513–V516.

Li, X., Jiang, H., 2005. A constrained joint optimization method for large margin HMM estimation. In: Proceedings of 2005 IEEE workshop on Automatic Speech Recognition and Understanding.

Li, X., 2005. Large Margin Hidden Markov Models for Speech Recognition. M.S. Thesis, Department of Computer Science and Engineering, York University, Canada.

Li, X., Jiang, H., 2006. Solving large margin HMM estimation via semi-definite programming. In: Proceedings of 2006 International Conference on Spoken Language Processing (ICSLP'2006), Pittsburgh, USA.

Li, X., Jiang, H., 2007. Solving large margin hidden Markov model estimation via semidefinite programming. IEEE Transactions on Audio, Speech and Language Processing 15 (8), 2383–2392.

Liu, C.-J., Jiang, H., Li, X., 2005. Discriminative training of CDHMMs for Maximum relative separation margin. In: Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2005), Philadelphia, Pennsylvania, pp. V101–V104.

Liu, C.-J., Jiang, H., Rigazio, L., 2005. Maximum relative margin estimation of HMMs based on *N*-best string models for continuous speech recognition. In: Proceedings of IEEE workshop on Automatic Speech Recognition and Understanding.

Liu, C., Liu, P., Jiang, H., Soong, F., Wang, R.-H., 2007. A constrained line search optimization for discriminative training in speech recognition. In: Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2007), Hawaii, USA.

Liu, P., Liu, C., Jiang, H., Soong, F., Wang, R.-H., 2008. A constrained line search optimization method for discriminative training of HMMs. IEEE Transactions on Audio, Speech and Language Processing 16 (5), 900–909.

Macherey, W., Haferkamp, L., Schluter, R., Ney, R., 2005. Investigations on error minimizing training criteria for discriminative training in automatic speech recognition. In: Proceedings of Interspeech, Lisbon, Portugal, pp. 2133–2136.

McDermott, E., Hazen, T.J., 2004. Minimum classification error training of landmark models for real-time continuous speech recognition. In: Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2004), pp. I-937–I-940.
McDermott, E., Hazen, T., Le Roux, J., Nakamura, A., Katagiri, S., 2007. Discriminative training for large-vocabulary speech recognition using minimum classification error. IEEE Transactions on Audio, Speech, and Language Processing 15 (1), 203–223.
Meir, R., 1995. Empirical risk minimization versus maximum likelihood estimation: a case study. Neural Computation 7 (1), 144–157.
Nadas, A., 1983. A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. IEEE Transactions on Acoustics, Speech and Signal Processing 31 (4), 814–817.
Nadas, A., Nahamoo, D., Picheny, M.A., 1988. On a model-robust training method for speech recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 36 (9), 1432–1436.
Neal, R., Hinton, G.E., 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan, M.I. (Ed.), Learning in Graphical Models. Kluwer Academic Publishers, pp. 355–368.
Normandin, Y., Cardin, R., Demori, R., 1994. High-performance connected digit recognition using maximum mutual information estimation. IEEE Transactions on Speech and Audio Processing 2 (2).
Pan, Z., Jiang, H., 2008. Large margin multinomial mixture model for text categorization. In: Proceedings of Interspeech 2008, Brisbane, Australia.
Pan, Z., 2008. Large margin multinomial model for document classification, Master Thesis, Department of Computer Science and Engineering, York University, Toronto, Canada.
Povey, D., Woodland, P.C., 2002. Minimum phone error and I-smoothing for improved discriminative training. In: Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2002), Orlando.
Povey, D., 2004. Discriminative training for large vocabulary speech recognition, Ph.D. Dissertation, Cambridge University, Cambridge, UK.
Rao, A.V., Rose, K., 2001. Deterministically Annealed Design of Hidden Markov Model Speech Recognizers. IEEE Transactions on Speech and Audio Processing 9 (2), 111–126.
Rose, K., 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proceedings of the IEEE 86 (11), 2210–2239.
Scholkopf, B., Smola, A.J., 2002. Learning with Kernels: Support vector Machine, Regularization, Optimization, and Beyond. The MIT Press, Cambridge.
Schluter, R., 2000. Investigations on discriminative training criteria, Ph.D. Dissertation, RWTH Aachenm University Technology, Aachen, Germany.
Smola, A.J., Bartlett, P., Scholkopf, B., Schuurmans, D., (Ed.), 2000. Advances in Large Margin Classifiers, The MIT Press.
Taskar, B., Guestrin, C., Koller, D., 2003. Max-margin Markov networks. In: Proceedings of Neural Information Processing Systems Conference (NIPS03), Vancouver, Canada.
Valtchev, V., 1995. Discriminative methods for HMM-based speech recognition, Ph.D. Dissertation, Cambridge University, UK.
Vapnik, V.N., 1998. Statistical Learning Theory. Wiley.
Wessel, F., Schluter, R., Macherey, K., Ney, H., 2001. Confidence measures for large vocabulary continuous speech recognition. IEEE Transactions on Speech and Audio Processing 9 (3), 288–298.
Woodland, P.C., Povey, D., 2002. Large scale discriminative training of hidden Markov models for speech recognition. Computer Speech & Language 16 (1), 25–47.
Yin, Y., Jiang, H., 2007. A fast optimization method for large margin estimation of HMMs based on second order cone programming. In: Proceedings of Interspeech 2007.
Yin, Y., 2007. A study of convex optimization for discriminative training of hidden Markov models in automatic speech recognition, Master Thesis, Department of Computer Science and Engineering, York University, Toronto, Canada.
Yin, Y., Jiang, H., 2007. A compact semidefinite programming (SDP) formulation for large margin estimation of HMMs in speech recognition. In: Proceedings of 2007 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Kyoto, Japan.

Further reading

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.
Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.
McDermott, E., Katagiri, S., 2004. A derivation of minimum classification error from the theoretical classification risk using Parzen estimation. Computer Speech and Language 18, 107–122.
Schluter, R., Macherey, W., Muller, B., Ney, H., 2001. Comparison of discriminative training criteria and optimization methods for speech recognition. Speech Communication 34, 287–310.