An Overview of Discriminative Training for Speech Recognition

Keith Vertanen
Computer Speech, Text and Internet Technology
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD United Kingdom

Abstract

This paper gives an overview of discriminative training as it pertains to the speech recognition problem. The basic theory of discriminative training will be discussed and an explanation of maximum mutual information (MMI) given. Common problems inherent to discriminative training will be explored as well as practicalities associated with implementing discriminative training for large vocabulary recognition. Alternatives to the MMI objective function such as minimum word error (MWE) and minimum phone error (MPE) will be discussed. The application of discriminative techniques for adaptation will be described. Finally, possible future avenues of research will be given.

1. Introduction

The accuracy of modern state-of-the-art speech recognition software relies on HMMs with properly trained parameters. Historically, the predominant training technique has been maximum likelihood estimation (MLE). MLE attempts to maximize the likelihood of the training data observations $O = \{O_1, O_2, .., O_R\}$:

$$F_{MLE}(\lambda) = \sum_{r=1}^{R} \log P_\lambda(O_r \mid M_{w_r}) \quad (1)$$

where λ is the set of model parameters and M_{w_r} is the HMM corresponding to the transcription w_r of observation O_r.

The popularity of MLE is due to its ability to produce accurate systems that can be quickly trained using the globally convergent Baum-Welch algorithm [1]. MLE also offers the theoretical advantage that if certain modeling assumptions hold, no other training criteria will do better; MLE is a minimum variance, consistent estimator of the true model parameters [2]. MLE makes a number of assumptions: observations are from a known family of distributions (typically Gaussian), training data is unlimited, and the true language model is known. Unfortunately, in general none of these assumptions hold for speech.

Given that MLE's assumptions are false, it is not guaranteed to produce optimal results. This has led researchers to explore whether better performing system could be constructed using other techniques such as discriminative training. Discriminative training attempts to optimize the correctness of a model by formulating an objective function that in some way penalizes parameter sets that are liable to confuse correct and incorrect answers.

In this paper, we will focus on the application of discriminative training as it pertains to the speech recognition problem. In section two, the theoretical basis for the dominant discriminative training objective function maximum mutual information (MMI) will be given and some of the problems associated with using it discussed. In section three, the

practicalities of implementing discriminative training will be examined, including the use of the extended Baum-Welch algorithm and word lattices. Section four will explore ways in which the generalization of discriminative training can be improved. In section five, several alternative objective functions to MMI will be described. Section six will describe how discriminative criteria can be used in speaker adaptation and adaptive training. Finally in section seven, finding will be summarized and possible future avenues of research given.

2. Maximum Mutual Information

2.1. Information Theoretic Basis

The basis for MMI can best be appreciated when approached from an information theoretic standpoint as in Brown [2]. Given the observation sequence O, a speech recognizer should choose a word sequence W such that there is a minimal amount of uncertainty about the correct answer. In other words, we want to minimize the conditional entropy:

$$H(W \mid O) = -\sum_{w,o} P(W = w, O = o) \log P(W = w \mid O = o) \qquad (2)$$

From (2), it is easy to see that by minimizing the conditional entropy, the probability of the word sequence given the observation must increase. This is intuitively what we want, an engine that makes good guesses.

The mutual information I(W;O) between W and O, can be written as:

$$I(W;O) = H(W) - H(W \mid O) \equiv$$
$$H(W|O) = H(W) - I(W;O) \qquad (3)$$

Thus if our goal is to minimize H(W|O), then we can try and minimize H(W) or maximize I(W;O). The minimization of H(W) corresponds to finding a language model with minimum entropy. This is a difficult problem as the probabilities of all possible word sequences for a natural language recognizer are not known and must be estimated. Most research focuses on the maximization of the mutual information term instead.

2.2. Maximum Mutual Information Estimation (MMIE)

Using the expressions for entropy, conditional entropy, and (3), it can be shown [2] that maximizing the mutual information on a set of observations O, requires choosing the parameter set λ to maximize the function:

$$F_{MMIE}(\lambda) = \sum_{r=1}^{R} \log \frac{P_\lambda(O_r \mid M_{w_r}) P(w_r)}{\sum_{\hat{w}} P_\lambda(O_r \mid M_{\hat{w}}) P(\hat{w})} \qquad (4)$$

where M_w is the HMM corresponding to the transcription w, P(w) is the probability of the word sequence w as determined by the language model, and the denominator sums over each possible word sequences $\hat{w}$.

To maximize (4), the numerator must be increased while the denominator is decreased. The first term in the numerator is identical to the objective function for MLE. Just like MLE, MMIE will try and maximize the likelihood of each observation given the training transcriptions. The difference in MMIE is the denominator term which can be made small by reducing the probabilities of other possible word sequences. Thus MMIE attempts to both make the correct hypothesis more probable, while at the same time making incorrect hypotheses less probable.

For certain simple types of estimation problems, it has been shown that give incorrect modeling assumptions, MMIE will converge to an optimal solution while MLE will not [2, 12]. However, it is also possible to construct problems in which neither MMIE or MLE will converge to an optimal solution [13]. As pointed out in [5], MMIE's robustness to model incorrectness on toy problems does not necessarily indicate it will be robust for real problems. MMIE's utility relies on how well it performs in practice.

2.3. Problems with MMIE

MMIE, along with other forms of discriminative training, have three main problems:

- Difficult to maximize objective function
- Computationally expensive to maximize objective function
- Poor generalization to unseen data

The objective functions in discriminative training cannot be optimized using the conventional Baum-Welch algorithm. The only known methods that converge for MMIE are steepest gradient descent and the extended Baum-Welch algorithm [4]. Given the high dimensionality of the parameter space, gradient descent may require a large number of iterations to obtain an optimal solution [3]. Thus extended Baum-Welch is the predominant algorithm used for parameter re-estimation in discriminative training. The details of extended Baum-Welch will be discussed in the next section.

The expense for computing the MMIE objective function stems from the denominator in (4). The denominator requires a summation over all possible word sequences in the language model. This amounts to performing recognition on each training utterance and for each iteration of the training algorithm. This might be possible when training a small vocabulary recognizer, but it quickly becomes computationally intractable for large vocabulary recognition. As we will see in the next section, approximations can be made that make the denominator computation tractable.

Finally, discriminative training techniques often perform very well on the training data, but fail to generalize well to unseen test data. This generalization failure may result from the domination in the numerator and denominator of (4) by a very few number of paths [1] or from the modeling of training data features which are not globally significant to the task [3]. Various techniques for addressing the generalization problem will be covered in section four.

3. Implementation

3.1. Discriminative Updating of HMMs

Mean and Variance Updates

The objective functions used in discriminative training such as (4) are rational functions. The original Baum-Welch algorithm requires that the objective function be a homogenous polynomial. In [4], the Baum-Welch algorithm was extended for use on rational functions. Further work in [5], allowed the use of extended Baum-Welch (EBW) on continuous density HMM systems utilizing a discrete approximation to a Gaussian.

The re-estimation formulas for the mean and diagonal covariance matrices for a state j and mixture component m are [5]:

$$\hat{\mu}_{jm} = \frac{\{\theta_{jm}^{num}(O) - \theta_{jm}^{den}(O)\} + D\mu_{jm}}{\{\gamma_{jm}^{num} - \gamma_{jm}^{den}\} + D} \quad (5)$$

$$\hat{\sigma}_{jm}^2 = \frac{\{\theta_{jm}^{num}(O^2) - \theta_{jm}^{den}(O^2)\} + D(\sigma_{jm}^2 + \mu_{jm}^2)}{\{\gamma_{jm}^{num} - \gamma_{jm}^{den}\} + D} - \hat{\mu}_{jm}^2 \quad (6)$$

where $\theta_{jm}(\cdot)$ is the sum of all observations or squared observations weighted by the probability of being in state j and mixture component m:

$$\theta_{jm}(O) = \sum_{r=1}^{R}\sum_{t=1}^{T_r} O^r(t)\gamma_{jm}^r(t) \quad (7)$$

$$\theta_{jm}(O^2) = \sum_{r=1}^{R}\sum_{t=1}^{T_r} [O^r(t)]^2\gamma_{jm}^r(t) \quad (8)$$

where O is the set of training utterances $\{O^1, O^2, \ldots, O^R\}$, each utterance O^i consists of observations $\{O^i(1), O^i(2), \ldots, O^i(T_i)\}$ and $\gamma_{jm}^r(t)$ is the probability of being in state j and component m at time t in observation r.

In (5) and (6), γ_{jm} is the sum over all time of the probability of being in state j and component m. The num and den superscripts indicate whether the summations are over the numerator or denominator model. The D variables are constants whose values will be discussed later in this section.

Note that (6) applies only to diagonal covariance matrices. The formula can be extended to full covariance matrices as shown in [1]. Experimentally full covariance matrices have been shown to improve results on continuous phoneme recognition [7]. The author is not aware of any recent experimental results on the use of full covariance matrices for large vocabulary tasks. This is probably due to an unacceptably large increase in parameters required by a full covariance matrix.

Mixture Weights and Transition Probability Updates

From the work in [4], re-estimation formula for mixture weights can be obtained in a similar manner. Unfortunately, this results in a formula that is extremely sensitive to small valued parameters. A more robust approximation was given in [9] and has been shown to work well in practice [7, 8]. This approximation also has a drawback in that it can lead to instability as training proceeds as the objective function may decrease as it approaches its maximum [8]. In [6], an update formula is proposed based on finding $\hat{c}_{jm}$ to maximize the auxiliary function:

$$\sum_{m=1}^{M} \gamma_{jm}^{num} \log \hat{c}_{jm} - \frac{\gamma_{jm}^{den}}{c_{jm}} \hat{c}_{jm} \quad (9)$$

where c_{jm} are the original weights, M is the number of mixture components in state j, and where (9) sums to one.

Similarly, the updating of transition probability can be done for each row of the transition matrix. Optimization of (9) can be done using a generic function optimization technique or by iteratively finding the optimal of one mixture weight while holding the other weights

constant. Happily, there are no constants to set for the mixture weight and transition probability update formulas.

The overall contribution made by mixture weights and transition probabilities is highly dependent on what type of sharing (if any) is being done between states or Gaussian mixture models. As discussed in [10], updating mixture weights and transition probabilities for a decision-tree tied-state mixture Gaussian HMM caused only a small increase in performance. Other HMM systems based on tied mixture models could expect a much greater effect.

Mixture Splitting

When designing a recognizer, a balance must be struck between the complexity of the HMM and the ability to train the parameters on a limited amount of data. Allocating a fixed number of Gaussians per state while a simple topology, is probably not an optimum use of our scarce parameter resources. One technique that has been shown to improve performance is allocating a varying number of Gaussians per state. Determining which states get additional Gaussians can be done using MLE, but can also be done using MMIE.

The derivative of the MMI objective function with respect to the mixture component weights can be used as an indicator of how much a particular Gaussian would benefit from being split [11]. Using this derivative, an algorithm can be devised which iteratively splits the top components which would most benefit from the discrimination that an additional Gaussian might provide. For a connected digit recognition task, MMIE-split mixture models averaging 2.5 Gaussians per state improved recognition by 41% relative over MLE-split models with 8 Gaussians per state [11].

For the harder WSJ task, discriminative mixture splitting using the same number of parameters as a model with a fixed number of Gaussians per state, improved performance by 3-5% relative [6]. Mixture splitting results were only given for a number of mixture components less than the optimal number given in [6]. It would be interesting to know how MMIE-based splitting performs as we approach the trainable limit of model parameters.

Setting the Smoothing Constant

The mean and variance update formulas (5) and (6) rely on the proper setting of the D smoothing constant. If D is too large, the step size is small and convergence is slow. If D is too small, the algorithm may become unstable. Preliminary work used a value of D that was the twice the minimum positive value needed to insure all variance updates were positive [5].

As can be seen from (5) and (6), the effect D has on the step side depends on the magnitude of the counts γ_{jm}. Frequently used models with large counts require a bigger value for D while infrequently used models need a smaller value. To account for this discrepancy, both state and Gaussian specific values for D have been proposed [6, 10].

In [10], using Gaussian specific D constants is reported to provide improved convergence speed over state specific constants (no specific numerical comparison was given). The Gaussian specific D_{jm} was set to twice the value required to insure positive variance updates for all Gaussian dimensions subject to a floor value of a global constant E multiplied by γ_{jm}^{den}. The constant E was set by first finding the minimum value for positive variance updates for

all Gaussians and setting E to half the maximum of $\frac{D_{jm}}{\gamma_{jm}^{den}}$ for all Gaussians. This setting of E experimentally produced the best test set performance and also avoided over-training that occurred when a fixed value of E was used.

3.2. Calculation of Statistics

Gaussian Occupancies and Observation Sums

The formulas given in 3.1 require the calculation of the Gaussian occupancies γ_{jm} (i.e. the probability of being in state j and component m summed over all time). They also require the sums $\theta_{jm}(\cdot)$ of all observations and squared observations weighted by the probability of being in state j and mixture component m.

In MLE training, these quantities are calculated for each training observation O_r using the forward-backward algorithm on the HMM M_{w_r} built from the transcription w_r. For MMIE training, we need not only the counts associated with the numerator of (4) but also for the denominator. The summation in the denominator of (4) is over HMMs built for each possible word sequence $\hat{w}$. We can calculate the occupancies during a full recognition pass on O_r using a composite HMM M_{den} built from all possible word sequences. For small vocabulary systems using small amounts of training data, this is tractable. But for large vocabulary systems using large amounts of training data, performing a full recognition pass on each training utterance for every training iteration is too computationally expensive.

Representing Hypotheses

We would like an alternate representation to the full recognition model that allows the approximate denominator occupancies to be cheaply computed. One possibility is to use N-best lists which are calculated once and used to constrain the search during subsequent MMIE iterations [14]. However, N-best lists are not good at representing the many possibilities that can be present in large vocabulary tasks. Lattice structures are a better candidate as they are able to succinctly represent the many hypotheses. While other types of lattices such as looped lattices have been investigated [15], we will focus on the word-based lattice.

A word lattice (see figure 1) consists of a number of nodes placed along the time axis of a particular training utterance. Arcs between nodes correspond to the recognition of a particular word. Arcs are also associated with the acoustic model and language model score for that word. The various paths through the word lattice represent the set of the most probable word sequences for a given model with a certain level of pruning.

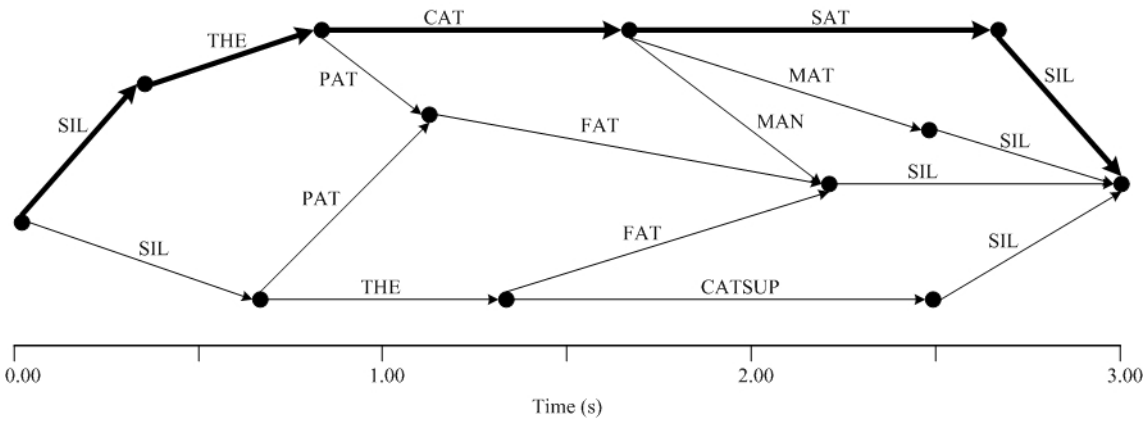

Figure 1: An example word lattice. The path in bold was the correct transcription of the utterance.

Lattice-Based Training

In [19], an explanation of the use of lattices for computing the denominator statistics is given. First, word lattices are constructed using the initial numerator and denominator MLE-trained HMMs for each of the training utterances. A lattice is used for the numerator to allow for multiple pronunciations of the same word. The denominator lattice is constructed using a pruning beam to control the size of resulting lattice. If the denominator lattice fails to contain a path corresponding to the training utterance transcription, the numerator lattice is merged into the denominator lattice. Typically, lattices are used repeatedly during each iteration of MMIE based on the assumption that the set of most probable word sequences does not change as training progresses. However, in some cases occasional lattice regeneration is beneficial (see section 5.2).

During each training iteration, the word lattices are used to create phone-marked lattices in which each word arc is labeled with the HMM model sequence and Viterbi segmentation points. Note that the acoustic and language models used to generate these labels may be different from the ones used to generate the initial word lattices. Using these phone-marked lattices, the forward-backward algorithm can be used to compute the statistics needed to re-estimate the model parameters. These new model parameters can then be used for the next iteration, a new set of phone-marked lattices created, and so on.

Using this lattice-based technique, MMIE training becomes tractable for large vocabulary tasks using a large amount of training data. Large scale experiments involving up to 265 hours of training data and using quinphone models, MMIE was shown to improve performance by 3.4% absolute over MLE on the Hub5 task [19].

4. Improving Generalization

4.1. Frame Discrimination

As discussed earlier, one of the main problems with discriminative training is its generalization to unseen test data. One approach to addressing this problem is to increase the amount of confusable data in the training set. This allows the training algorithm to learn more possible confusions with the hope that these confusions will be seen in the test data.

As shown in the example in figure 2, due to the network topology, it is not possible for the observations O_4, O_5, O_6 ("cat") to be used in the training of the models for "mat". It could also be the case that some of the paths in M_{den} were pruned from the word lattice. For example, the language model might predict that "the cat mat" is a very unlikely word sequence, preventing the "mat" model for being discriminated from the observations O_7, O_8, O_9 ("sat").

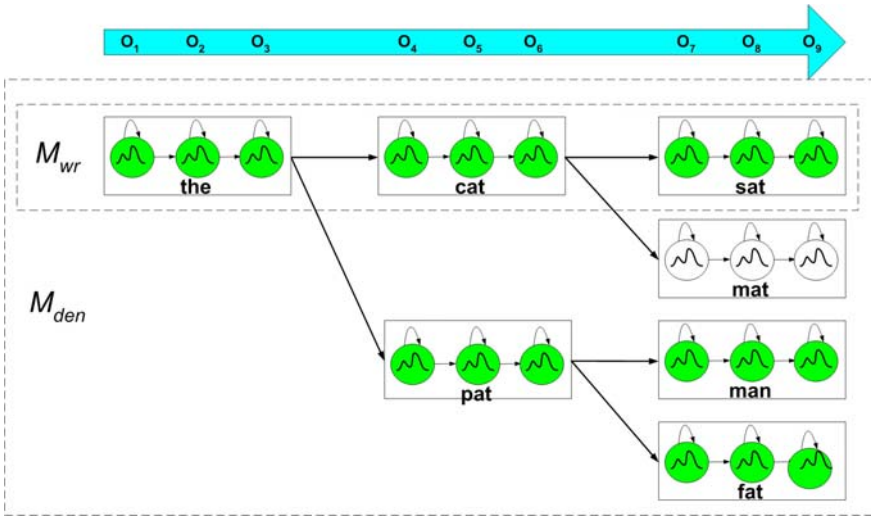

Figure 2: Training utterance (O_1, O_2, .., O_9) as applied to numerator and denominator models.

One way of increasing the amount of confusable data proposed in [1] is frame discrimination (FD). In FD, the M_{den} model is relaxed to allow more possible state sequences. The specific relaxed model explored in [1] is a zero memory Markov chain. As shown in figure 3, this can be thought of as placing all the Gaussians in parallel. Each observation has a chance to update any of the Gaussians in any of the states.

Experimentally frame discrimination has met with mixed results. The original author showed FD could improve MMIE results for digit recognition [1]. However, for harder tasks such as Resource Management, North American Business [16] and broadcast news transcription [17], FD did not significantly improve accuracy for complex models. Using clever selection of the best Gaussians, FD had been shown to speed computation, but this advantage has been supplanted by advancements in lattice-based techniques [17].

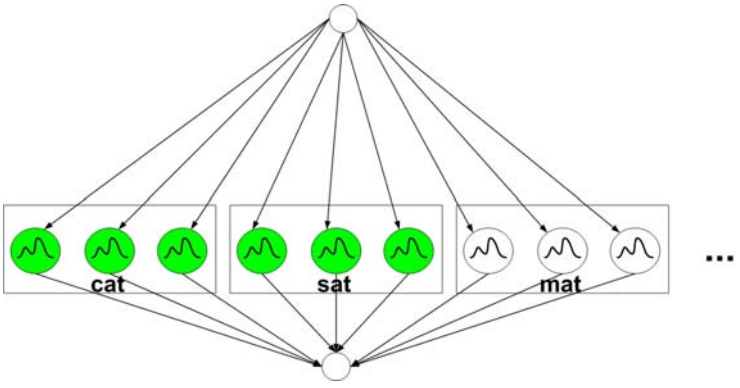

Figure 3: Frame discrimination can be thought of as putting all Gaussians in parallel.

4.2. Acoustic Model Scaling

In computing the probability of a word sequence, it is traditional to scale the language model log likelihood by a positive value. This adjusts for the underestimation in the probabilities of the acoustic model due to invalid modeling assumptions. As discussed in [19], for the purposes of discriminative training we want to scale the acoustic model instead. This allows more state sequences in the lattice to make contributions to the summations used to calculate the EBW update formulas. This was shown to increase the number of states with high posterior probability which should increase the amount of confusable states and thus improve generalization. This was bore out experimentally in [19] with acoustic scaling beating language scaling by 0.40% absolute on 18 hours of training data. A comparison of the effects of acoustic model scaling for larger amounts of training data was not reported.

4.3. Weakened Language Models

The language model used for the initial generation of the word lattices can be different from the language model used for recognition. For example, a unigram model might be used to build the lattice while a trigram is used for recognition. Using a weaker language model for lattice construction allows the lattice to contain more confusable state sequences which should hopefully improve generalization.

On the Wall Street Journal task, using a unigram for lattice generation and a trigram for recognition resulted in a 0.32% absolute increase in performance over using a trigram for lattice generation [18]. For the harder Hub5 task, it was shown in [19] that using a unigram for lattice generation and a trigram for recognition slightly improved performance over using a bigram for lattice generation. As no results in [19] were given for using a trigram for lattice generation and recognition, the overall impact of weakened language models could not be assessed.

4.4. H-criterion

Rather than increasing confusable data to improve generalization, another possibility is to use an objective function that is an interpolation between MMIE and MLE:

$$\alpha F_{MMIE} + (1-\alpha)F_{MLE} \tag{10}$$

This technique was investigated in [19] and found to improve test set performance using 68 hours of training data. However, choosing an optimal value of α is difficult and the effectiveness of this technique decreases with increasing amounts of training data. But even without accuracy gains, the H-criterion might be useful to help prevent over-training by making it less critical exactly when training is stopped.

4.5. I-smoothing

As the H-criterion failed to improve accuracy for large amounts of training data, a variant technique I-smoothing was proposed in [20]. I-smoothing increases the weight of the MLE counts depending on the amounts of data available for each Gaussian. This is done by multiplying the numerator terms γ_{jm}^{num}, $\theta_{jm}^{num}(O)$, and $\theta_{jm}^{num}(O^2)$ in the EBW formulas by:

$$1 + \frac{\tau}{\gamma_{jm}^{num}} \tag{11}$$

where τ is a constant. As τ increases from zero, more weight is given to MLE.

I-smoothing was shown to improve MMIE test-set performance on the Switchboard/Call Home task by 0.40% absolute using 265 hours of training data.

5. Alternate Objective Functions

Rather than maximizing the mutual information, we could try to directly minimize the errors made by the recognizer on the training set. The minimum classification error (MCE) objective function is designed to minimize these errors and have been shown to outperform MMIE on small tasks [21]. However, using MCE on large vocabulary tasks is problematic for long sentences and also cannot easily be implemented on lattices. In [20] the alternates minimum word error (MWE) and minimum phone error (MPE) were proposed.

5.1. Minimum Word Error (MWE)

The minimum word error (MWE) objective function attempts to minimize the number of word level errors made by maximizing:

$$F_{MWE}(\lambda) = \sum_{r=1}^{R} \log \frac{\sum_{\hat{w}} P_\lambda(O_r \mid M_{\hat{w}})^k \cdot P(\hat{w}) \cdot \text{RawAccuracy}(\hat{w})}{\sum_{\hat{w}} P_\lambda(O_r \mid M_{\hat{w}})^k \cdot P(\hat{w})} \quad (12)$$

where $\text{RawAccuracy}(\hat{w})$ measures the number of words that were correct in word sequence $\hat{w}$, k is the acoustic model scaling factor. Note that in (12) we are assuming that that acoustic and language model probabilities have not previously been scaled (in contrast to [20]). As discussed in section 4.2, for generalization reasons, acoustic model scaling is preferable to language model scaling.

The function (12) is the weighted average of the correct words in all possible sequences. By increasing (12), the number of correct words in the most probable sequences is increased. In order to efficiently calculate $\text{RawAccuracy}(\hat{w})$ in a lattice-based recognizer, an approximation is taken:

$$\text{RawAccuracy}(\hat{w}) = \sum_{w \in \hat{w}} \text{WordAcc}(w) \quad (13)$$

$$\text{WordAcc}(w) = \begin{cases} -1 + 2e & \text{if same word} \\ -1 + e & \text{if different word} \end{cases} \quad (14)$$

where e is the proportion that the word w overlaps in time with the word in the correct transcription.

5.2. Minimum Phone Error (MPE)

Instead of maximizing the word accuracy, we could instead maximize the phone level accuracy. The formulation of the minimum phone error (MPE) objective function is identical to (12) except $\text{RawAccuracy}(\hat{w})$ sums the number of correct phones. Calculating MPE statistics for each arc in the lattice, the EBW update formulas can then be employed to update the model parameters [23].

In [20], the use of the I-smoothing technique (section 4.5) was required in order for MWE or MPE to improve upon MMIE. While MWE was shown to give better training set performance, MPE results in better test set performance. On the Switchboard/Call Home task using 265 hours of training data, MPE improved upon an MMIE I-smoothed system by 1.0% absolute. Further improvements using MPE were shown in [22] by regenerating word lattice after several rounds of MPE training and merging the lattice with the original. This gave an additional 0.5% absolute improvement on the Switchboard task.

6. Discriminative Adaptation

6.1. Discriminative Speaker Adaptation

One common feature in state-of-the-art recognizers is the use of linear transforms to adapt speaker independent models to better approximate a particular speaker. The most popular technique is maximum likelihood linear regression (MLLR) [24]. As the name implies, MLLR uses the maximum likelihood principle and has been found to yield significant performance improvements.

One immediate concern is whether the gains made by MLLR on MLE trained models will continue to hold using discriminatively trained models. Woodland [19] shows that MLLR provided similar accuracy gains on MLE and MMIE trained models on the Hub5 task. This is contradicted by the findings of McDonough, et al. [25] where MLLR was shown to cause no improvement on MMI trained models on the English Spontaneous Scheduling Task (ESST). A second experiment using Broadcast News (BN) and ESST was conducted in [25], but results for MMI-MLLR were noticeably absent.

A discriminative linear transform (DLT) for adaptation can be arrived at by interpolating the ML and MMI objective functions using the H-criterion [26]. While the iterative formula given is not guaranteed to increase with each step, with a properly chosen H-criterion constant, it works well in practice. For the task of recognizing non-native speakers with a model trained on native speakers, DLT-based adaptation gave on average a 0.8% absolute improvement over standard MLLR.

6.2. Discriminative Speaker Adaptive Training

The training data from which speaker independent recognizers are trained is usually collected from a wide range of speakers. The inter-speaker variability is a possible source of recognizer error. In speaker adaptive training (SAT) [27], these differences are minimized by the application of transforms calculated on the training speakers.

In discriminative speaker adaptive training (DSAT), similar to the speaker adaptation case, DLTs are computed for the normalization of the training set speakers. Results in [28] show that using DSAT and MMI-trained models provides a 0.6% absolute increase over normal MMI on conversational telephone speech. Using MPE, a larger increase of 0.8% absolute is attained.

7. Conclusions

In this paper, an overview has been given of the fundamental theories and techniques used in discriminate training for speech recognition. We have seen why conventional MLE is lacking and explained the use of an alternate method MMIE which performs better in the face of model incorrectness.

Some of the problems inherent to discriminative training were discussed and possible solutions discussed. We saw that extended Baum-Welch algorithm provides a viable framework for the re-estimation of model parameters in discriminative training. The use of word lattices to constrain the denominator search was shown to make the calculations needed for EBW tractable. We saw how generalization to unseen test data could be improved by

techniques such as frame discrimination, acoustic model scaling, weakened language models, and smoothing with MLE.

We introduced the alternative objective functions MWE and MPE. These functions allow estimates for the error rate of a recognizer to be used directly in training. Discriminative training was shown to also allow improvements in both speaker adaptation and speaker adaptive training.

Through out this paper, we have seen how the various discriminative training methods have consistently outperformed traditional MLE training on both simple and complex recognition tasks.

Future Work

While there has been a lot of activity surrounding the use of discriminative techniques in speech recognition, there are still many interesting avenues of research. Some possibilities include:

- Dynamic lattice size
  As we have seen, the use of lattices is critical in making discriminative training tractable for large vocabulary tasks. This presumes that the lattices, due to pruning, are a significantly reduced search space during the training iterations. It is possible accuracy could be improved by dynamically adjusting pruning amount used on individual training utterances based on some measure of their recognition difficulty or acoustic confusability. This would allow more resources to be devoted to training utterances where there might be more to learn by the exploration of a larger hypothesis space.
- Adjustment of model topology
  A recognizer typically standardizes on one particular topology for its acoustic unit HMMs. For example, a common topology is a five state HMM with three emitting states and strict left-to-right transitions (including self-loops). It would be interesting if discriminative criteria could be developed that allowed models to adjust their topology based on the need to discriminate themselves from other models. For example, a particular phone might have a significantly longer duration distribution than other phones. This phone might benefit from additional HMM states.
- Discriminative training for language models
  There has been little investigation into the application of discriminative techniques to language modeling. In [29], modest gains were shown using a very small amount of training data. It would be interesting to know if similar techniques could provide gains for state-of-the-art language models trained on many megawords of training data.

8. References

[1] Kapadia, S. (1998). Discriminative Training of Hidden Markov Models. *Ph.D. thesis*, University of Cambridge.

[2] Brown, P. (1987). The Acoustic-Modelling Problem in Automatic Speech Recognition. *Ph.D. thesis*, Carnegie-Mellon University.

[3] Valtchev, V. (1995). Discriminative Methods in HMM-based Speech Recognition. *Ph.D. thesis*, University of Cambridge.

[4] Gopalakrishnan, P.S., Kanevsky, D., Ndas, A., Nahamoo, D. (1991). An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems. *IEEE Transactions on Information Theory*, Vol. 37, pp. 107-113.

[5] Normandin, Y., Morgera, D. (1991), An Improved MMIE Training Algorithm for Speaker-Independent Small Vocabulary, Continuous Speech Recognition. *Proceedings of the IEEE, ICASSP91*, pp. 537-540.

[6] Valtchev, V., Odell, J.J., Woodland, P. C., Young, S.J. (1997), MMIE Training of Large Vocabulary Recognition Systems, *Speech Communications*, Vol. 22, pp. 303-314.

[7] Kapadia, S., Valchev, V., Young, S.J. (1993). MMI Training for Continuous Phoneme Recognition on the TIMIT Database. *Proceedings of the IEEE, ICASSP93*, Vol. 2, pp. 491-494.

[8] Normandin, Y. (1991), Hidden Markov Models, Maximum Mutual Information Estimation and the Speech Recognition Problem. *PhD thesis*, McGill University.

[9] Merialdo, B. (1988), Phonetic Recognition Using Hidden Markov Models and Maximum Mutual Information Training. *Proceedings of the IEEE, ICASSP88,* Vol. 1, pp. 111-114.

[10] Woodland, P.C., Povey, D. (2002), Large Scale Discriminative Training of Hidden Markov Models for Speech Recognition. *Computer Speech and Language*, Vol. 16, pp. 25-47.

[11] Normandin, Y. (1995), Optimal Splitting of HMM Gaussian Mixture Components with MMIE Training. *Proceedings of the IEEE, ICASSP95*, Vol. 1, pp. 449-452.

[12] Nádas, A., Nahamoo, D., Picheny, M.A (1988), On a Model-Robust Training Method for Speech Recognition, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-36, pp. 1432-1436.

[13] Gopallakrishnan, P.S., Kanevsky, D., Nádas, A., Nahamoo, D. (1989), A Generalization of the Baum Algorithm to Rational Objective Functions, *Proceedings of the IEEE, ICASSP89*, paper S12.9.

[14] Chow, Y.L. (1990), Maximum Mutual Information Estimation of HMM Parameters for Continuous Speech Recognition Using the N-Best Algorithm. *Proceedings of the IEEE, ICASSP90*.

[15] Normandin, Y., Lacouture, R., Cardin, R. (1994), MMIE Training for Large Vocabulary Continuous Speech Recognition. *Proceedings of the IEEE, ICASLP94*, pp. 1367-1371.

[16] Povey, D., Woodland, P.C. (2000), Frame Discrimination of HMMs for Large Vocabulary Speech Recognition. *Technical Report CUED/F-INFENG/TR.232*, Cambridge University Engineering Department.

[17] Woodland, P.C., Hain, T., Moore, G.L., Niesler, T.R., Povey, D., Tuerk, A., Whittaker, E.W.D. (1999), The 1998 HTK Broadcast News Transcription System: Development and Results. *Proceedings DARPA Broadcast News Workshop*, pp. 265-270.

[18] Schlüter, R., Müller, B., Wessel, F., Ney, H. (1999), Interdependence of Language Models and Discriminative Training. *Proceedings of the IEEE ASRU Workshop*, pp. 119-122.

[19] Woodland, P.C., Povey, D. (2000), Large Scale Discriminative Training for Speech Recognition. *Proceedings of International Workshop on Automatic Speech Recognition*.

[20] Povey, D., Woodland, P.C. (2002), Minimum Phone Error and I-Smoothing for Improved Discriminative Training. *Proceedings of the IEEE, ICASSP02,* Orlando.

[21] Juang, B., Chou, W., Lee, C. (1997), Minimum Classification Error Rate Method for Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 5.

[22] Woodland, P, Chan, R., Evermann, G., Gales, M., Hain, T., Kim, D., Liu, A., Mrva, D., Povey, D., Tranter, S., Wang, L, Yu, K. (2003), 2003 CU-HTK English CTS Systems. *2003 Rich Transcription Workshop*.

[23] Woodland, P., Evermann, G., Gales, M., Hain, T., Liu, A., Povey, D., Wang, L. (2002), CU-HTK April 2002 Switchboard System. *2002 Rich Transcription Workshop*.

[24] Leggetter, C.J., Woodland, P. (1995), Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMMs. *Computer Speech & Language,* Vol. 9, pp. 171-186.

[25] McDonough, J., Schaaf, T., Waibel, A. (2002), On Maximum Mutual Information Speaker-Adapted Training, *Proceedings of the IEEE, ICASSP02.*

[26] Uebel, L.F., Woodland, P.C. (2001), Discriminative Linear Transforms for Speaker Adaptation, *Proceedings ITRW ASR.*

[27] Anastasakos, J., McDonough, J., Schwarz, R., Makhoul, J. (1996), A Compact Model for Speaker-Adaptive Training, Proceedings ICSLP, pp. 1137-1140.

[28] Wang, L., Woodland P.C. (2003), Discriminative Adaptation and Adaptive Training, *EARS STT Workshop*.

[29] Kuo, H.K., Fosler-Lussier, E., Jiang, H., Lee, C.H. (2002), Discriminative Training of Language Models for Speech Recognition, *Proceedings of the IEEE, ICASSP02.*