FMPE: DISCRIMINATIVELY TRAINED FEATURES FOR SPEECH RECOGNITION

Daniel Povey, Brian Kingsbury, Lidia Mangu, George Saon, Hagen Soltau, Geoffrey Zweig

IBM T.J. Watson Research Center, NY; {dpovey,bedk,mangu,gsaon,hsoltau,gzweig}@us.ibm.com

ABSTRACT

MPE (Minimum Phone Error) is a previously introduced technique for discriminative training of HMM parameters. fMPE applies the same objective function to the features, transforming the data with a kernel-like method and training millions of parameters, comparable to the size of the acoustic model. Despite the large number of parameters, fMPE is robust to over-training. The method is to train a matrix projecting from posteriors of Gaussians to a normal size feature space, and then to add the projected features to normal features such as PLP. The matrix is trained from a zero start using a linear method. Sparsity of posteriors ensures speed in both training and test time. The technique gives similar improvements to MPE (around 10% relative). MPE on top of fMPE results in error rates up to 6.5% relative better than MPE alone, or more if multiple layers of transform are trained.

1. INTRODUCTION

This article introduces fMPE, a method of discriminatively training features. The MPE objective function is reviewed in Section 2; Sections 3 and 4 describe fMPE; Section 5 discusses some issues relating to its use; experiments are presented in Sections 7 and 6, and conclusions are presented in Section 8.

2. MINIMUM PHONE ERROR (MPE)

The Minimum Phone Error (MPE) objective function for discriminative training of acoustic models was previously described in [1, 2]. The basic notion is the same as other discriminative objective functions such as MMI, i.e. training the acoustic parameters by forcing the acoustic model to recognize the training data correctly.

The MPE criterion is an average of the transcription accuracies of all possible sentences s, weighted by the probability of s given the model:

$$\mathcal{F}_{\mathrm{MPE}}(\lambda) = \textstyle\sum_{r=1}^{R} \sum_{s} P_{\lambda}^{\kappa}(s|\mathcal{O}_r)\mathrm{A}(s, s_r) \qquad (1)$$

where $P_{\lambda}^{\kappa}(s|\mathcal{O}_r)$ is defined as the scaled posterior sentence probability $\frac{p_{\lambda}(\mathcal{O}_r|s)^{\kappa}P(s)^{\kappa}}{\sum_u p_{\lambda}(\mathcal{O}_r|u)^{\kappa}P(u)^{\kappa}}$ of the hypothesized sentence s, where λ is the model parameters and $\mathcal{O}_r$ the r'th file of acoustic data.

The function $\mathrm{A}(s, s_r)$ is a "raw phone accuracy" of s given s_r, which equals the number of phones in the reference transcription s_r for file r, minus the number of phone errors.

3. FMPE

3.1. High-dimensional feature generation

The first stage of fMPE is to transform the features into a very high dimensional space. A set of Gaussians is created by likelihood-based clustering of the Gaussians in the acoustic model to an appropriate size (up to 100,000 in experiments reported here). On each frame, the Gaussian likelihoods are evaluated with no priors, and a vector of posteriors is formed. This can be done very quickly (e.g. less than 0.1xRT) by further clustering the Gaussians to, say, 2000 cluster centers and only evaluating the 100 most likely clusters based on the cluster-center's likelihood [3].

3.2. Acoustic context expansion

The vector is further expanded with left and right acoustic context. The following is a typical configuration used: If the central (current) frame is at position 0, vectors are appended which are the average of the posterior vector at positions 1 and 2, at positions 3, 4 and 5, and at positions 6, 7, 8 and 9. The same is done to the left (positions -1 and -2, etc) so that the final vector is of size 700,000 if there were 100,000 Gaussians. Sparse vector routines are used for speed.

3.3. Feature projection

The high dimensional features are projected down to the dimension of the original features $\mathbf{x}_t$ and added to them, so

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{M}\mathbf{h}_t \qquad (2)$$

i.e. the new feature $\mathbf{y}_t$ equals the old features plus the high-dimensional feature $\mathbf{h}_t$ obtained as described above, times a matrix $\mathbf{M}$. Initializing $\mathbf{M}$ to zero gives a reasonable starting point for training, i.e. the original features.

3.4. Training the matrix

The matrix is trained by linear methods, because in such high dimensions accumulating squared statistics would be impractical. The update on each iteration is:

$$M_{ij} := M_{ij} + \nu_{ij} \frac{\partial \mathcal{F}}{\partial M_{ij}}, \qquad (3)$$

i.e. gradient descent where the parameter-specific learning rates are:

$$\nu_{ij} = \frac{\sigma_i}{E(p_{ij} + n_{ij})}, \quad (4)$$

where p_{ij} and n_{ij} (see below) are the sum over time of the positive and negative contributions towards $\frac{\partial \mathcal{F}}{\partial M_{ij}}$, E is a constant that controls the overall learning rate and σ_i is the average standard deviation of Gaussians in the current HMM set in that dimension. Since $\frac{\partial \mathcal{F}}{\partial M_{ij}} = p_{ij} - n_{ij}$, the most each M_{ij} can change is $1/E$ standard deviations, and the most any given feature element y_{ti} can change is n/E standard deviations, where n is the number of acoustic contexts by which the vector H_t has been expanded (e.g. n =7).

It follows from Equation 2 that

$$\frac{\partial \mathcal{F}}{\partial M_{ij}} = \sum_{t=1}^{T} \frac{\partial \mathcal{F}}{\partial y_{ti}} h_{tj}, \quad (5)$$

where h_{tj} is the j'th dimension of $\mathbf{h}_t$ and y_{ti} is the i'th dimension of the transformed feature vector $\mathbf{y}_t$. The differential $\frac{\partial \mathcal{F}}{\partial M_{ij}}$ is broken into the positive and negative parts needed to set the learning rate in Equation 4:

$$p_{ij} = \textstyle\sum_{t=1}^{T} \max(\frac{\partial \mathcal{F}}{\partial y_{ti}} h_{tj}, 0) \quad (6)$$

$$n_{ij} = \textstyle\sum_{t=1}^{T} \max(-\frac{\partial \mathcal{F}}{\partial y_{ti}} h_{tj}, 0). \quad (7)$$

3.5. Smoothing of update

To prevent over-training of parameters that cannot be estimated robustly, a modification is made as follows. Let the "count" c_{ij} be $\sum_{t=1}^{T} h_{tj}$, which is similar to the number of nonzero points available in estimating the differential $\frac{\partial \mathcal{F}}{\partial M_{ij}}$. This formula only makes sense if the high dimensional features h_{tj} are generally either zero or not far from one; another way to set c_{ij} is $(\sum_{t=1}^{T} |d_{ij}(t)|)^2 / \sum_{t=1}^{T} d_{ij}(t)^2$ where $d_{ij}(t) = \frac{\partial \mathcal{F}}{\partial y_{ti}} h_{tj}$, which is the number of points that would have the same expected ratio of squared sum of absolute values to sum-of-squares if it were Gaussian distributed with zero mean. These approaches gives similar counts. The count c_{ij} is used to work out the typical magnitude of a nonzero differential which is $(p_{ij} + n_{ij})/c_{ij}$. This is used to "pad" the differentials p_{ij} and n_{ij} with a number τ of typical imaginary observations prior to update, so $n_{ij} := n_{ij} + 0.5\tau(p_{ij} + n_{ij})/c_{ij}$, and $p_{ij} := p_{ij} + 0.5\tau(p_{ij} + n_{ij})/c_{ij}$. This slows down the learning rate (Equation 4) for parameters that have too few observations. Smoothing may slightly improve results, on the order of 0.1% absolute; generally this is done with $\tau \simeq 100$.

Some experiments reported here pad the two statistics with imaginary counts that are not equal, but have the same ratio as the overall statistics for the relevant cluster of Gaussians. However this does not make any clear difference to the WER so it is not described further.

4. CALCULATING THE DIFFERENTIAL

4.1. Direct differential

As mentioned in Section 3.4, a key quantity in fMPE training is $\frac{\partial \mathcal{F}}{\partial y_{ti}}$ which is the differential of the MPE function w.r.t. the i'th dimension of the transformed feature vector on time t.

Directly differentiating the MPE objective function can be done via the following equation. Defining the log likelihood of Gaussian m of state s on time t as l_{smt},

$$\frac{\partial \mathcal{F}}{\partial y_{ti}}^{\text{direct}} = \textstyle\sum_{s=1}^{S} \sum_{m=1}^{M_s} \frac{\partial \mathcal{F}}{\partial l_{smt}} \frac{\partial l_{smt}}{\partial y_{ti}}. \quad (8)$$

The first factor $\frac{\partial \mathcal{F}}{\partial l_{smt}}$ is already calculated in normal MPE training [1, 2]; it equals $\sum_{q=1}^{Q} \kappa \gamma_q^{\text{MPE}} \gamma_{qsm}(t)$ where κ is the probability scale, $\kappa \gamma_q^{\text{MPE}}$ is the differential of $\mathcal{F}$ w.r.t. the log likelihood of the q'th phone arc, and $\gamma_{qsm}(t)$ is the Gaussian occupation probability within the phone arc. The second factor $\frac{\partial l_{smt}}{\partial y_{ti}}$ equals $\frac{\mu_{smi} - y_{ti}}{\sigma_{smi}^2}$. Note that the positive and negative γ_q^{MPE} (and the positive and negative l_{smt}) should sum to zero on each time t, and if for numerical or pruning reasons they do not it may be wise to re-balance the statistics arising from the positive and negative parts.

4.2. Indirect differential

Equation 8 is unsatisfactory because it takes no account of the fact that the same features are used to train as well as test the model, and the features will affect the HMM parameters. When using Equation 8 for the differential, it was found that much of the WER improvement was lost as soon as the same features were used to to retrain the models (with ML training). For this reason, the differential is augmented with a term that reflects changes in the models. The statistics used for normal MPE training are used to calculate $\frac{\partial \mathcal{F}}{\partial \mu_{smi}}$ and $\frac{\partial \mathcal{F}}{\partial \sigma_{smi}^2}$, i.e. the differential of the objective function w.r.t. the model means and variances (see Section 4.3). This allows us to calculate the part of the differential that is mediated by changes in the Gaussians:

$$\frac{\partial \mathcal{F}}{\partial y_{ti}}^{\text{indirect}} = \quad (9)$$

$$\textstyle\sum_{s=1}^{S} \sum_{m=1}^{M_s} \frac{\gamma_{sm}(t)}{\gamma_{sm}} \left(\frac{\partial \mathcal{F}}{\partial \mu_{smi}} + 2 \frac{\partial \mathcal{F}}{\partial \sigma_{smi}^2} (y_{ti} - \mu_{smi}) \right)$$

where $\gamma_{sm}(t)$ is the ML occupation probability as used in standard forward-backward training; γ_{sm} is the same thing summed over all the training data. The final differential that is used is:

$$\frac{\partial \mathcal{F}}{\partial y_{ti}} = \frac{\partial \mathcal{F}}{\partial y_{ti}}^{\text{direct}} + \frac{\partial \mathcal{F}}{\partial y_{ti}}^{\text{indirect}}. \quad (10)$$

Note that Equation 9 is based on assumptions that are not quite met. The fMPE differential of Equation 8 and the MPE differentials $\frac{\partial \mathcal{F}}{\partial \mu_{smi}}$ etc are the differentials around the current acoustic parameters and features. The current acoustic parameters λ were generated from statistics obtained by aligning previous models, say λ^{prev}. Ideally, Equation 9 should refer to these previously obtained occupation probabilities $\gamma_{sm}(t)^{\text{prev}}$ and $\gamma_{sm}^{\text{prev}}$. For convenience this is not done.

4.3. Model parameter differentials

In order to calculate the indirect differential, the quantities $\frac{\partial \mathcal{F}}{\partial \mu_{smi}}$ and $\frac{\partial \mathcal{F}}{\partial \sigma^2_{smi}}$ are are obtained from normal MPE statistics [1, 2] as follows:

$$\frac{\partial \mathcal{F}}{\partial \mu_{smi}} = \frac{\kappa}{\sigma^2_{smi}} \left(\theta^{\text{num}}_{smi}(\mathcal{O}) - \theta^{\text{den}}_{smi}(\mathcal{O}) - \mu_{smi}(\gamma^{\text{num}}_{sm} - \gamma^{\text{den}}_{sm}) \right), \tag{11}$$

where μ_{smi} and σ^2_{smi} are the mean and variance in the Gaussians used for the alignment, and $\theta^{\text{num}}_{smi}(\mathcal{O})$ and $\gamma^{\text{num}}_{smi}$ etc are the sum-of-data and count MPE statistics.

For the variance, let us first define the quantities S^{num}_{smi} and S^{den}_{smi} which are the variance of the numerator and denominator statistics around the current mean, so e.g.

$$S^{\text{num}}_{smi} = (\theta^{\text{num}}_{smi}(\mathcal{O}^2) - 2\theta^{\text{num}}_{smi}(\mathcal{O})\mu_{smi} + \gamma^{\text{num}}_{sm}\mu^2_{smi})/\gamma^{\text{num}}_{sm}, \tag{12}$$

where $\theta^{\text{num}}_{smi}(\mathcal{O}^2)$ are the sum-of-squared-data statistics. The differential w.r.t the variance is then

$$\frac{\partial \mathcal{F}}{\partial \sigma^2_{smi}} = \frac{\kappa\gamma^{\text{num}}_{sm}}{2}(S^{\text{num}}_{smi}\sigma^{-4}_{smi} - \sigma^{-2}_{smi}) - \frac{\kappa\gamma^{\text{den}}_{sm}}{2}(S^{\text{den}}_{smi}\sigma^{-4}_{smi} - \sigma^{-2}_{smi}). \tag{13}$$

4.4. Checks

A useful check that no implementation errors have been made is that adding a small quantity to all the features in some dimension should not affect the MPE objective function, as long as it is done in both training and test. This implies that

$$\sum_{t=1}^{T} \frac{\partial \mathcal{F}}{\partial y_{ti}}^{\text{direct}} + \frac{\partial \mathcal{F}}{\partial y_{ti}}^{\text{indirect}} = 0, \tag{14}$$

where the summation $\sum_{t=1}^{T}$ is over all training data. The two terms in the above equation generally cancel out to within a margin of, say 1% of the absolute values of the two terms. Discrepancies are due to the assumptions made in Equation 9 not being met. A similar metric relating to a linear scaling of each dimension can be more sensitive to problems but should cancel to within a few percent:

$$\sum_{t=1}^{T} y_{ti}\frac{\partial \mathcal{F}}{\partial y_{ti}}^{\text{direct}} + y_{ti}\frac{\partial \mathcal{F}}{\partial y_{ti}}^{\text{indirect}} = 0. \tag{15}$$

5. OVERVIEW AND GENERAL CONSIDERATIONS IN FMPE TRAINING

5.1. Overview

Procedurally, each iteration of fMPE training involves three passes over the data: one to accumulate normal MPE statistics; a second to accumulate fMPE statistics (chiefly the quantities n_{ij} and p_{ij}), and a third pass to do an ML update with the newly transformed data. All three passes start with the same HMMs; for simplicity, in these experiments the third pass aligns with the newly transformed features rather than doing single-pass retraining from the old to the new features. Naturally, on the $n+1$'th iteration the updated HMMs from the n'th iteration will be used to align the data and the first two passes will use the transformed features from the n'th iteration. Convergence speed is similar to MPE, so three or four iterations may give the best performance.

5.2. Dimension of high-dimensional features

Experiments on call center data suggest that it is probably good to use as high a dimension as possible until there is insufficient data for each parameter and data-learning becomes an issue. This is why the very high dimension of 100,000 × 7 contexts was used in CTS experiments reported here. The overhead in testing is very small - about 0.1 to 0.2xRT. Much of the improvement in WER can be obtained with a smaller dimension and no acoustic context. Early experiments used state posteriors rather than Gaussian posteriors; no clear evidence is available as to their relative usefulness but Gaussian posteriors are more convenient.

5.3. Typical criterion improvements

In fMPE, the improvement in MPE criterion (expressed relative to the number of phones in the correct transcription) tends to be smaller than in MPE training: around 2-3% absolute, e.g. rising from 0.70 to 0.725, compared with perhaps 6% in MPE training. However the observed WER improvements on test data are not much smaller than the criterion improvement (say, around 2%); also in fMPE training a greater proportion of the training data criterion improvement is seen when the MPE criterion is measured on unseen data, as compared with MPE training. Note that the MPE criterion is a kind of smoothed error rate so the comparison with WER makes sense.

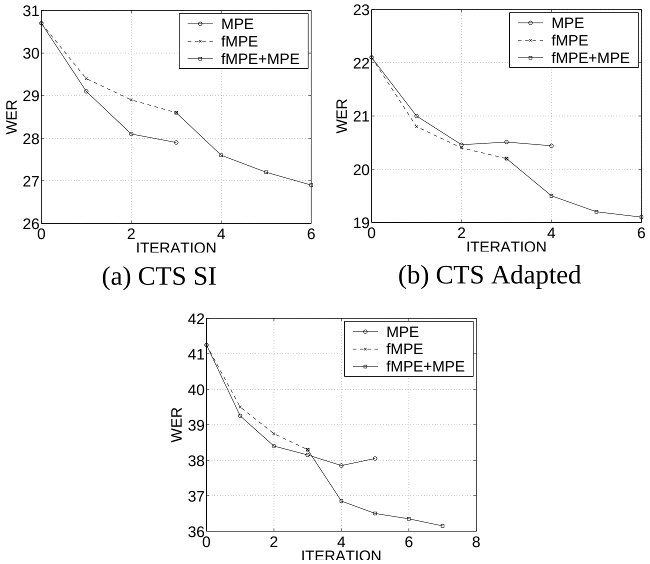
5.4. Typical learning rates, and acoustic scaling

The values of E used in the CTS experiments reported here are 0.96 for the speaker independent system, and 1.44 for the speaker adapted system (which had 7 acoustic contexts in the high dimensional features, vs. 5 in the speaker independent (SI) system). The call-center experiments also use 7 contexts and $E = 1.44$. For the best values of E (in terms of WER on test data), the proportion of parameters M_{ij} that changes sign seems to be around 10-15% on the second iteration, decreasing to around 5-10% on subsequent iterations; the average absolute values of the M_{ij} that change sign is around 1/4 that of those that do not. The predicted MPE criterion improvement based on $\frac{\partial \mathcal{F}}{\partial M_{ij}}$ and the change in M_{ij} tends to be around 6% to 12% (0.06 to 0.12) on the first iteration, decreasing to half that or less on the second.

To prevent the fMPE transform from attempting to generally strengthen or weaken the acoustic model relative to the LM, the differential of the MPE criterion w.r.t a scaling of all the acoustic likelihoods was calculated and the LM weight was tuned until this was close to zero. The speaker adapted CTS system, for example, had $k = 0.1$ (acoustic weight) and an LM weight of 1.25.

6. CONVERSATIONAL TELEPHONE SPEECH (CTS) EXPERIMENTS

The setup for MPE is largely as described in [1]; however a fourth set of statistics (corresponding to the denominator statistics in MMI training) is also accumulated so that



(a) CTS SI

(b) CTS Adapted

(c) Call-center SI

Fig. 1. MPE and fMPE results

I-smoothing can back off to an MMI rather than an ML estimate. The lattices for the speaker independent (SI) experiments use a unigram LM; those for the adapted experiments use a highly pruned bigram LM (150k bigrams). In adapted experiments the statistics are averaged over several acoustic and LM scales (0.10 and 0.16 acoustic, and 1.0 and 1.6 LM; four combinations); there is weak evidence that this works well when combined with a bigram language model. Variances are floored to the 20th percentile of the cumulative distribution of variances in each dimension [2].

In Figure 1(a) and (b), results for MPE training and fMPE followed by MPE are shown on the NIST conversational telephone speech (CTS) task in both SI and adapted conditions; these experiments were done in preparation for IBM's submission to the NIST RT-04 (Rich Transcription 2004) evaluation [4]. Training is on 2300h of telephone speech data. Both systems used cross-word phonetic context, and PLP features with LDA+MLLT projections to 40 dimensions (SI) and 39 (adapted). Testing is on RT-03.

The SI system is a quinphone system with 8k states and 150k Gaussians. The high-dimensional features are posteriors of 64k clustered Gaussians with five contexts (a subset of the contexts described in Section 3.2). The transform is trained with 1/5 of the training data. As shown in Figure 1(a), fMPE+MPE is better by 1.0% than MPE alone.

The adapted system has 7-phone context, 22k states and 850k Gaussians, training and testing on VTLN+fMLLR features. The $\mathbf{h}_t$ are posteriors of 100k Gaussians, with seven contexts (700k dimensions total). The transform is trained on all the data. In this case fMPE alone is better than MPE alone, perhaps because MPE does not work well with very large acoustic models. The final fMPE+MPE number, at 19.1%, is better by 1.3% than MPE alone.

For the RT-04 evaluation, a system with 0.4% better WER than the final fMPE+MPE number was obtained. Do do this, the fMPE features were used to train from scratch a small 5-phone context system. Then, a second layer of fMPE transform ("iterated fMPE") was trained on the small system using 1/4 the data, with 25k Gaussians $\times$ 7 contexts. This doubly transformed data was used to further train the original 7-phone context fMPE models (20.2% $\rightarrow$ 19.4%), after which MPE training was done ($\rightarrow$ 18.7%). This is 1.7% better than the best models with MPE alone. The final transcriptions submitted included other features such as cross-adaptation, MLLR, LM rescoring and consensus. The 10xRT system had 13.0% WER on Dev-04, and 16.1% on RT-03 with 12.4% on the Fisher portion only.

7. CALL CENTER EXPERIMENTS

Figure 1(c) shows experiments on data recorded from an IBM computer support call center. No adaptation is used. Training is on 300h of speech; the models have 11-phone left phonetic context, 4k states and 97k Gaussians. Test data is 6 hours long. Features are PLP projected with LDA+MLLT to 40 dimensions. High dimensional features are 32k Gaussian posteriors with 7 contexts (224,000 dimensions). MPE is with backoff to MMI as above. The fMPE+MPE results on call-center data are an impressive 5.1% better than the ML baseline and 1.7% better than MPE alone.

8. CONCLUSION

fMPE is a novel and effective way to apply discriminative training to features rather than models. This makes possible things that are not possible with normal discriminative training, such as building a system on the new features and iterating the process. It made a significant contribution to IBM's submission to the RT-04 evaluation.

9. REFERENCES

[1] Povey D. and Woodland P.C., "Minimum Phone Error and I-smoothing for Improved Discriminative Training," in *ICASSP*, 2002.

[2] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University, 2004.

[3] Saon G., Zweig G., Kingsbury B., Mangu L., and Chaudhari U., "An Architecture for Rapid Decoding of Large Vocabulary Conversational Speech," in *Eurospeech*, 2002.

[4] Soltau H., Kingsbury B., Mangu L., Povey D., Saon G., and Zweig G., "The IBM 2004 Conversational Telephony System for Rich Transcription in EARS," in *ICASSP*, 2005.