Recent Progress on the Discriminative Region-dependent Transform for Speech Feature Extraction

Bing Zhang[†]*, Spyros Matsoukas, Richard Schwartz*

BBN Technologies
50 Moulton St, Cambridge, MA, 02138
{bzhang,smatsouk,schwartz}@bbn.com

Abstract

The region-dependent transform (RDT) is a feature extraction method for speech recognition that employs the Minimum Phoneme Error (MPE) criterion to optimize a set of feature transforms, each concentrating on a region of the acoustic space. Previous results have shown that RDT gives significant recognition-error reduction in a large vocabulary speaker-independent (SI) system. As a follow-up investigation, this paper presents the recent progress of applying RDT in speaker-adaptive training (SAT). Similar to previous SI results, the integration of RDT with SAT yields 7% relative improvement in word error rate (WER). Also, theoretical comparisons are made between RDT and other discriminative feature extraction methods, including the improved version of the feature-space MPE (fMPE) that uses the "mean-offsets" as additional input features.

Index Terms: speech recognition, discriminative training, feature extraction, region-dependent transform.

1. Introduction

In speech recognition, the acoustic features are often extracted from the cepstral coefficients using linear transforms, which can be estimated using LDA[1], HLDA [2], or MLLT [3]. However, the criteria of these methods do not correlate well with the recognition errors, hence the features optimized under such criteria are not optimal in terms of minimizing WER.

The discriminative region-dependent transform (RDT) is developed to overcome this problem by using the MPE [4] criterion that is closely related to WER. In RDT, the acoustic space is divided into multiple regions through a global Gaussian mixture (GMM) model. Each region is asssociated with a distinct feature transform. In particular, we use the linear projection of long span features as the regional transform. At run-time, given any observation, the posterior probabilities of the Gaussians are used to determine which region the observation belongs to, hence what transform to apply. An advantage of RDT is that large number of parameters and nonlinearity can be introduced without too much computational cost.

RDT is related to several discriminative feature training methods. MMI-SPLICE [5] also uses the GMM to divide the acoustic space into regions, but it only has one bias vector associated with each region as a correction term of existing features. This scheme can be viewed as a special case of RDT. Feature-space MPE (fMPE) [6] has a different origin from SPLICE, since it treats the posteriors of the Gaussians in the GMM as candidate features, and uses a linear projection to reduce the dimensionality. The projected vectors are used as correction terms to existing features, so that the linear projection can be simply initialized with a zero matrix. The core of fMPE is equivalent to SPLICE after mathematical rearrangement [7]. An improvement to fMPE is introduced in [8], where the differences between the original feature and Gaussian means, so-called mean-offsets, are treated as features in addition to the posteriors. It is interesting that the "mean-offset" fMPE, developed from a totally different perspective, can be written as a form of RDT. This will be analyzed in Section 2.

In our original paper [9], using RDT in the SI English Conversational Telephone Speech (CTS) system offered about 6% relative WER reduction. Although it is possible to use the SI-RDT in SAT, the fear is that the ML-based speaker adaption may compromise the gain obtained from the discriminative feature optimization. In order to solve this problem, we have developed a procedure to integrate RDT with SAT, where RDT is estimated under the presence of the speaker-dependent (SD) transforms. Using this procedure, referred to as SA-RDT (speaker adaptive RDT), we have obtained a similar gain to that of the SI-RDT.

The paper is organized as follows. Section 2 gives a review of RDT, as well as its relation to other methods. The experimental part is presented in section 3, where we show the procedure of SA-RDT, and summarize the results of both SI-RDT and SA-RDT with the English CTS systems.

2. Region-dependent Transform

The region-dependent feature transform was introduced as a method for discriminative feature extraction [9]. It uses a global Gaussian mixture model (GMM) to divide the acoustic space into multiple regions, each having a different transform. The output feature of RDT is a weighted average of the region-specific features, defined as

$$x_t = F_{\text{RDT}}(o_t) = \sum_{i=1}^{N} \kappa_t^{(i)} f_i(o_t) \qquad (1)$$

with o_t the vector of input features at time t, N being the total number of regions, $\kappa_t^{(i)}$ being the posterior probability of Gaussian i given o_t, and f_i being a region-dependent parametric vector-to-vector function.

In order to include information from the acoustic context, RDT uses long-span features as input, which are obtained by concatenating several frames of PLP [10] coefficients. Specifically, if c_t is used to denote the PLP vector at time t, the long-span feature

[†] Bing Zhang is a Ph.D. student at the College of Computer & Information Science, Northeastern University.

vector o_t will contain coefficients from time $t-\ell$ to $t+\ell$:

$$o_t = [c_{t-\ell}^T, ..., c_{t-1}^T, c_t^T, c_{t+1}^T, ..., c_{t+\ell}^T]^T \quad (2)$$

As a special case of RDT, the regional function f_i is a linear transform:

$$f_i(o_t) = A_i o_t + b_i \quad (3)$$

A_i is a projection used to select low-dimensional features from o_t, and b_i is a correction term in the projected space. This scheme is referred to as the region-dependent linear transform (RDLT).

2.1. RDLT vs. fMPE and mean-offset fMPE

In RDLT, if no parameter-sharing is used among the projections, distinct parameters in A_i will easily outnumber those in b_i, making the latter less noticeable. For this reason, we simply ignore the biases in our experiments, using only one projection for each region.

In the original fMPE [6], an opposite strategy is adopted, i.e., only one bias vector is used for each region [7, 9]. Since a bias vector does not have a lot of parameters, fMPE uses very large number of regions to increase the overall power of the transform.

Mean-offset fMPE [8] is a recent improvement to the original fMPE. It uses "mean-offset" features as well as Gaussian posteriors as input features of a global linear projection. It also employs an improved context expansion layer to apply weighted average of the features from adjacent frames in order to form the final feature vector of the current frame. Without its context expansion layer, mean-offset fMPE can be written as:

$$F_{\text{m-fMPE}}(x_t) = x_t + Mh_t \quad (4)$$

where the input feature x_t is a low-dimensional vector that is usually obtained by applying a fixed projection on the long-span feature vector o_t. We can assume that $x_t = Po_t$. h_t is a joint vector of a Gaussian posterior vector κ_t and a mean-offset vector δ_t.

$$h_t = [\eta\kappa_t^T, \delta_t^T]^T \quad (5)$$

$$\delta_t = [\kappa_t^{(1)} d_{(t,1)}^T, \kappa_t^{(2)} d_{(t,2)}^T, ..., \kappa_t^{(N)} d_{(t,N)}^T]^T \quad (6)$$

$$d_{(t,i)} = \Sigma_i^{-1}(x_t - \mu_i), \quad \forall i \in [1, N] \quad (7)$$

The mean-offset feature is the difference between the feature vector and the mean μ_i of Gaussian i, weighted by the inverse of Σ_i, the diagonal standard deviation matrix of Gaussian i. Mean-offsets are further weighted by the posteriors of Gaussians, and the posterior vector is scaled by a predetermined constant η, set to 5.0 in [8]. It is easy to see that if x_t has p dimensions, δ_t is a $N \cdot p$ dimensional vector, and h_t is a $N \cdot (p+1)$ dimensional vector.

From the perspective of fMPE, the input feature vector h_t is used to encode the position information of x_t. When mean-offsets are used, they provide additional information about the relative location of a frame within one Gaussian. In this way, using a GMM of a moderate size is probably enough to tell where an observation vector is.

Interestingly, mean-offset fMPE can be rewritten in terms of RDLT in spite of their different origins. First, let us rewrite Eq. (4) by breaking matrix M into two parts M_a and M_b:

$$F_{\text{m-fMPE}}(x_t) = x_t + M_a\delta_t + M_b\kappa_t \quad (8)$$

in which M_a is a $p \times N \cdot p$ matrix applied on mean-offset features, and M_b is the original $p \times N$ fMPE matrix, which is applied to the posteriors. For simplicity, we assume that the scalar η is included in M_b.

Now, if we break the columns of M_a and M_b into N groups respectively by Gaussian, and use the fact that $\sum_{i=1}^N \kappa_t^{(i)} = 1$, it follows that

$$\begin{aligned} F_{\text{m-fMPE}}(x_t) &= x_t + \sum_{i=1}^N (\kappa_t^{(i)} M_a^{(i)} d_{(t,i)} + \kappa_t^{(i)} M_b^{(i)}) \\ &= \sum_{i=1}^N \kappa_t^{(i)} \left[(I + M_a^{(i)}\Sigma_i^{-1})x_t + (M_b^{(i)} - M_a^{(i)}\Sigma_i^{-1}\mu_i) \right] \end{aligned} \quad (9)$$

where $M_a^{(i)}$ is the i-th $p \times p$ block of M_a, and $M_b^{(i)}$ is the i-th column vector of M_b.

Recall that $x_t = Po_t$. We can define a constrained RDLT as

$$F_{\text{c-RDLT}}(o_t) = \sum_{i=1}^N \kappa_t^{(i)} (C_i P o_t + b_i) \quad (10)$$

where C_i is a full-rank matrix. Obviously it is at least as general as the transform in Eq. (9). In fact, the two are equivalent to each other since the following equation array has a unique solution for all $M_a^{(i)}$ and $M_b^{(i)}$ given any C_i and b_i.

$$\begin{cases} C_i &= I + M_a^{(i)}\Sigma_i^{-1} & \forall i \in [1, N] \\ b_i &= M_b^{(i)} - M_a^{(i)}\Sigma_i^{-1}\mu_i & \forall i \in [1, N] \end{cases} \quad (11)$$

This observation explains why mean-offset fMPE is able to use less Gaussians than the original fMPE: simply because it has a more powerful feature transform in each region. It also suggests that there is nothing special about the mean-offset features.

Further more, in this analysis, we found that without its context expansion layer, mean-offset fMPE would be equivalent to a constrained RDLT, where the region-dependent transform happens in a projected space. In this case, anything rejected by the projection P (usually not discriminatively trained and suboptimal) will never be seen by the discriminative feature transform. In contrast to this, the unconstrained RDLT can reselect features from the acoustic context because it optimizes linear projections of long-span features. By this means, using an additional context expansion layer in mean-offset fMPE seems more necessary than in the unconstrained RDLT.

2.2. RDT optimization

RDT is optimized discriminatively under the MPE criterion, so that the features are more suited for recognition error reduction.

For R training utterances with transformed features $\mathbf{X} = \{X_1, ..., X_R\}$ and reference transcriptions $\{W_1^{\text{ref}}, ..., W_R^{\text{ref}}\}$, the MPE objective function [4] is defined as

$$\mathcal{H}_{\text{MPE}}(\mathbf{X}, \lambda) = \sum_r^R \sum_{W_r} \frac{p(X_r \mid W_r, \lambda)^\beta p(W_r) \alpha(W_r, W_r^{\text{ref}})}{\sum_{W_r'} p(X_r \mid W_r', \lambda)^\beta p(W_r')} \quad (12)$$

where $X_r = \{x_1^r, ..., x_T^r\}$ is the sequence of transformed feature vectors of utterance r, W_r is a hypothesis word sequence of the utterance, $\alpha(W_r, W_r^{\text{ref}})$ is the phoneme accuracy score of that hypothesis with respect to the reference, and $\lambda = \{\mu_1, ..., \mu_S, \Sigma_1, ..., \Sigma_S\}$ is the set of Gaussian means and covariances of an HMM that is trained from $\mathbf{X}$. The exponent β is used to reduce the dynamic range of the acoustic scores.

Although our goal is to optimize the features, how the HMM is trained affects the MPE derivative, hence the performance of the optimized feature transform, because λ depends indirectly on RDT through the transformed features $\mathbf{X}$. It is often preferable to assume ML update of the HMM rather than MPE update, so that the optimization will concentrate more on finding discriminative features rather than adapting features to a discriminative model.

The chain rule can be used to compute the derivative of Eq. (12) with respect to the parameters of RDT. For example, if (3) is the regional function of RDT, the derivatives are

$$\frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial A_i} = \sum_{r,t} \kappa_{r,t}^{(i)} \frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial x_t^r} o_t^{rT} \tag{13}$$

$$\frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial b_i} = \sum_{r,t} \kappa_{r,t}^{(i)} \frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial x_t^r} \tag{14}$$

The derivative of MPE with respect to x_t^r can be expressed as a summation of two terms:

$$\frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial x_t^r} = \left(\frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial x_t^r} \right)_\lambda + \frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial \lambda} \frac{\partial \lambda}{\partial x_t^r} \tag{15}$$

The first term is derived by fixing λ, and the second term is obtained via the chain rule.

If word lattices are used to represent the hypotheses, the first term of (15) can be computed via lattice-based forward/backward algorithm using the formula

$$\left(\frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial x_t^r} \right)_\lambda = -\beta \sum_{q \in Q_t^r} \sum_{m \in M_q} \mathcal{D}(q,m,t) \Sigma_m^{-1} (x_t^r - \mu_m) \tag{16}$$

The accumulation takes place in those arcs Q_t^r that contain x_t^r, with M_q being the Gaussians in arc q, and $\mathcal{D}(q,m,t)$ being a weighting scalar:

$$\mathcal{D}(q,m,t) = p(q \mid X_r, \lambda) \left[s_{\lambda,q}^r - s_\lambda^r \right] \psi_m^q(t) \tag{17}$$

First, $p(q \mid X_r, \lambda)$ is the posterior of the arc in the lattice. Secondly, the two scores $s_{\lambda,q}^r$ and s_λ^r are the expected accuracy score over all hypotheses that contain arc q_r, and the expected accuracy over all hypotheses (i.e., the MPE score of this utterance), respectively. Finally $\psi_m^q(t)$ is the posterior probability of Gaussian m within arc q.

The second term of the MPE derivative (15) depends on how the model is updated. When the model is updated via ML training, a forward/backward pass on the reference transcripts is needed in order to compute this term, using the equation below:

$$\frac{\partial \mathcal{H}_{\text{MPE}}(\mathbf{X},\lambda)}{\partial \lambda} \frac{\partial \lambda}{\partial x_t^r} = \beta \sum_m \frac{\gamma_m^r(t)}{\sum_{t'} \gamma_m^r(t')} \cdot \left[(x_t^r - \mu_m)(\Sigma_m^{-1} \mathcal{G}_m \Sigma_m^{-1} - \phi_m \Sigma_m^{-1}) + \Sigma_m^{-1} \mathcal{J}_m \right] \tag{18}$$

In (18), $\gamma_m^r(t)$ is the posterior probability of Gaussian m at time t given the observations and reference word sequence. Note that we have made an assumption that $\gamma_m^r(t)$ is independent of the current model λ, because otherwise the derivative would be too complicated. The assumption is true if we use a fixed model $\lambda^{(0)}$ and fixed features $\mathbf{X}^{(0)}$ to compute $\gamma_m^r(t)$. Similarly, λ should be updated via single-pass retraining (SPR) [11], in which $\lambda^{(0)}$ and $\mathbf{X}^{(0)}$ are used to compute the forward/backward variables. In practice, we find that the regular EM update also works, using the current model and features in the forward/backward pass, although in this case convergence cannot be guaranteed.

Finally, $\mathcal{G}_m$, $\mathcal{J}_m$ and ϕ_m are Gaussian-dependent statistics that should be accumulated over all data from the lattice-based forward/backward pass. They have forms similar to (16) and can be pre-computed in the lattice-based forward/backward pass.

$$\mathcal{G}_m = \beta \sum_{r,t} \sum_{q \in Q_t^r} \sum_{m \in M_q} \mathcal{D}(q,m,t)(x_t^r - \mu_m)(x_t^r - \mu_m)^T \tag{19}$$

$$\mathcal{J}_m = \beta \sum_{r,t} \sum_{q \in Q_t^r} \sum_{m \in M_q} \mathcal{D}(q,m,t)(x_t^r - \mu_m) \tag{20}$$

$$\phi_m = \beta \sum_{r,t} \sum_{q \in Q_t^r} \sum_{m \in M_q} \mathcal{D}(q,m,t) \tag{21}$$

We use a gradient descent algorithm [12] to update the parameters of RDT. λ can be updated through SPR, or approximately through the regular EM training.

3. Experiments

In [9], we evaluated the performance of RDT in an SI system that was trained on a 2300-hour EARS RT04 CTS training corpus. Testing was performed on the union of two sets: the RT03 evaluation set (Eval03), consisting of 3 hours of Switchboard-II and 3 hours of Fisher data, and the RT04 development set (Dev04), consisting of 3 hours of Fisher data.

The baseline system used a Vocal Tract Length Normalized (VTLN) PLP front-end, computing 14 cepstral coefficients and normalized energy per frame of speech (25 msec window length, 10 msec frame step), followed by mean and covariance normalization. The actual 60-dimensional features used in acoustic model training were produced by applying LDA+MLLT on sets of 15 contiguous cepstral frames (225 dimensions).

Recognition was carried out in three passes, using a composite within-word triphone State Tied Mixture (STM) HMM, a within-word quinphone State Clustered Tied Mixture (SCTM) model, and a crossword quinphone SCTM model, respectively. Refer to [9] for details of the language model and the decoding procedures.

The baseline crossword SCTM model consisted of approximately $900K$ Gaussians in $7K$ state clusters. MPE training with an MMI prior [13] was applied to the baseline models on unigram lattices, generated on the 2300-hour corpus using the ML models.

A 1000-region SI-RDT was initialized from the crossword LDA+MLLT projection. A smaller crossword SCTM model (44 Gaussian per state, $7K$ state clusters) was used to train the feature transform on the same lattices as in baseline MPE training. The final ML and MPE systems were trained using optimized RDT features.

The table below shows that SI RDT gives 9.3% and 5.8% relative WER reductions compared to the ML and MPE baselines, respectively.

Transform	ML Model WER	MPE Model WER
LDA+MLLT	22.5	20.4
SI-RDT	20.4	19.2

Table 1: Unadapted Eval03+Dev04 decoding results of ML and MPE models

Given the success of SI-RDT, a further question is whether RDT performs well in SAT procedures such as CMLLR-SAT [3] or HLDA-SAT [14]. In CMLLR-SAT, SD transforms are applied on top of the global transform to reduce the inter-speaker variability. In HLDA-SAT, another set of SD transforms are applied at the front-end to the cepstral coefficients and normalized energy.

A straightforward approach is to perform SAT on top of a model trained with SI-RDT. However, the problem is that when RDT is estimated, there is no consideration for the SD transforms that are applied on top of it. Such SD transforms, usually estimated under the ML criterion, could take away some of the gain from the discriminatively trained RDT.

Ideally, to integrate RDT with SAT, we should assume that λ in Eq. (12) is updated by SAT. However, the derivative would be too complicated to compute. Instead, we use the following procedure to integrate RDT with CMLLR-SAT, using alternating updates of the RDT and the SD transforms:

1. Train SI-RDT without SD transforms.
2. Train an SI crossword SCTM model using SI-RDT
3. Estimate SD transforms using the model from step 2.
4. Re-estimate RDT (referred to as SA-RDT) under the presence of fixed SD transforms. The derivative of MPE should be propagated through these transforms.
5. Use SA-RDT and the SD transform to finish the SAT, training an STM model and an SCTM non-crossword model (the SCTM crossword model has already been updated together with SA-RDT).

The integration of RDT with HLDA-SAT is not much different, because another set of SD transforms in HLDA-SAT is applied before RDT, which can be viewed as a kind of front-end normalization of cepstral coefficients.

The CMLLR-SAT experiments on English CTS use the same training and testing data, front-end analysis, LM, state clusters, and lattices as the SI experiments. The table below compares the performance of SA-RDT to the baseline system, and to the system trained with SI-RDT using the straightforward approach.

Transform	ML Model WER	MPE Model WER
LDA+MLLT	20.2	18.5
SI-RDT	18.8	17.6
SA-RDT	18.0	17.2

Table 2: Adapted Eval03+Dev04 decoding results of SAT-ML and SAT-MPE models

As we can see, better results are obtained by re-estimating RDT with fixed SD transforms. The relative gains over the baseline are similar to what we had on SI systems, measured as 10.9% and 7.0% for the ML and MPE systems, respectively.

4. Conclusions

As a complement to [9], this paper has presented a more theoretical analysis of RDT and experimental results on speaker adapted systems. We have shown that mean-offset fMPE, a substantially different method, has an core equivalent to a constrained version of RDT. We also give the formulae of the MPE derivative with respect to RDT in an implementation-oriented way, showing necessary training passes and sufficient statistics to accumulate. Finally, we have presented a procedure that performs alternating updates of the SA-RDT and the speaker-dependent transforms, which yields a 7% relative WER reduction in the English CTS SAT-MPE system. It is more effective than simply performing regular SAT on top of an SI-RDT.

5. Acknowledgment

This work was supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

6. References

[1] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience, 2nd edition, 2000.

[2] N. Kumar and A. G. Andreou, "A generalization of linear discriminant analysis in maximum likelihood framework," Tech. Rep. JHU-CLSP Technical Report No. 16, Johns Hopkins University, Aug. 1996.

[3] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," Tech. Rep. 291, Cambridge University, 1997.

[4] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proceedings of ICASSP*, Orlando, FL, May 2002, IEEE.

[5] J. Droppo and A. Acero, "Maximum mutual information SPLICE transform for seen and unseen conditions," In *Proceedings of Interspeech* [15].

[6] D. Povey, "fMPE: discriminatively trained features for speech recognition," in *Proceedings of ICASSP*, Philadelphia, PA, Mar. 2005, IEEE, pp. 961–964.

[7] L. Deng, J. Wu, J. Droppo, and A. Acero, "Analysis and comparison of two speech feature extraction/compensation algorithms," *IEEE Signal Processing Letters*, vol. 12, no. 6, Jun 2005.

[8] D. Povey, "Improvements to fMPE for discriminative training of features," In *Proceedings of Interspeech* [15].

[9] B. Zhang, S. Matsoukas, and R. Schwartz, "Discriminatively trained region dependent transforms for speech recognition," in *Proceedings of ICASSP*, Toulouse, France, May 2006, IEEE.

[10] H. Hermansky, "Perceptual linear predictive (PLP) analysis for speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[11] M. J. F. Gales, *Model-Based Techniques for Noise Robust Speech Recognition*, Ph.D. thesis, Cambridge University, 1995.

[12] D. C. Liu and J. Nocedal, "On the limited memory BFGS methods for large scale optimization," *Mathematical Programming 45*, pp. 503–528, 1989.

[13] D. Povey et al., "EARS Progress Update," presentation in EARS STT meeting, Nov. 2003.

[14] S. Matsoukas and R. Schwartz, "Improved speaker adaptation using speaker dependent feature projections," in *Proceedings of ASRU*, Virgin Islands, U.S., Nov. 2003, IEEE, pp. 273–278.

[15] ISCA, *Proceedings of Interspeech*, Lisbon, Portugal, Sept. 2005.