STREAMLINING THE FRONT END OF A SPEECH RECOGNIZER

Hua Yu and Alex Waibel

Interactive Systems Lab, Carnegie Mellon University, Pittsburgh, PA 15213
Email: hyu@cs.cmu.edu

ABSTRACT

In this paper we seek to streamline various operations within the front end of a speech recognizer, both to reduce unnecessary computation and to simplify the conceptual framework. First, a novel view of the front end in terms of linear transformations is presented. Then we study the invariance property of recognition performance with respect to linear transformations (LT) at the front end. Analysis reveals that several LT steps can be consolidated into a single LT, which effectively eliminates the Discrete Cosine Transform (DCT) step, part of the traditional MFCC (Mel-Frequency Cepstral Coefficient) front end. Moreover, a highly simplified, data-driven front-end scheme is proposed as a direct generalization of this idea. The new setup has no Mel-scale filtering, another part of the MFCC front end. Experimental results show a 5% relative improvement on the Broadcast News task.

1. LINEAR TRANSFORMATIONS IN THE TRADITIONAL FRONT END

The front end is a relatively independent component of a speech recognition system. Although the actual acoustic model parameters depend directly upon front-end parameterization, researchers tend to view it as a black box. When testing several different front ends, the acoustic model structure is seldom altered: it is simply a matter of plugging in another front end, re-estimating model parameters, and finally choosing the one that yields the lowest WER (Word Error Rate).

It is important to realize, however, that front-end design and acoustic modeling are closely coupled. Below we will go through a typical front end commonly seen in most LVCSR systems, with an emphasis on connections between the two components:

1. First, the Fourier spectrum is warped to compensate for gender/speaker differences (Vocal Tract Length Normalization, or VTLN).

2. The warped spectrum is then smoothed by integrating over triangular bins arranged along a non-linear scale. Mel-scale, the most commonly used one, is designed to approximate the frequency resolution of human ear, which is more sensitive at lower frequencies. Normally 30 triangular-shaped filters are used in JRTk (Janus Recognition Toolkit).

3. The log of the filter-bank output is taken to compress the dynamic range of the spectrum, so that the statistics of the estimated power spectrum are approximately Gaussian.

4. Next, cepstral coefficients are obtained by applying a Discrete Cosine Transform (DCT) to the log filter-bank outputs. The goal is mostly to achieve a decorrelation effect so that the subsequent modeling using diagonal covariance matrices is more valid. Typically, the first 13 coefficients are retained.

5. Cepstral Mean Normalization (CMN) is commonly used to normalize for the channel effect, so we can build a “channel-blind” acoustic model later.

6. Delta and double-delta features are appended to the MFCC vector to capture speech dynamics.

7. Finally, LDA (Linear Discriminant Analysis) can be used for dimensional reduction. On top of LDA, there can be a further diagonalization transform so that the feature vector fits better with the diagonal covariance assumption in the acoustic model [4, 3, 6] . This is also called Maximum Likelihood Linear Transform (MLLT), which happens to be a special case of semi-tied covariance matrices [2].

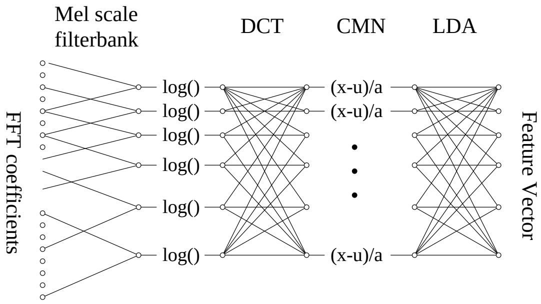

Figure 1: A Typical MFCC Front End [1]

[1] Here VTLN and Δ, $\Delta\Delta$ steps are not shown for simplicity.

It's easy to see that many of the operations in Figure 1 are linear transformations (LT). As a matter of fact, except for FFT and log, everything else is just a linear transformation. For example, Mel-scale filterbank is a matrix multiplied on FFT coefficients. And the same for DCT.

CMN is also linear. However, it differs from the others by the fact that it's not a global LT. It's either utterance based or cluster based, meaning that cepstral mean is estimated and subtracted per utterance/cluster, whereas other LTs are global (condition/class independent).

The simplest delta and double delta feature can be considered as a linear transformation over the extended vector:

$$\begin{pmatrix} x \\ \Delta_x \\ \Delta\Delta_x \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 - x_0 \\ x_2 - 2x_1 + x_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}$$

Here $x_0, x_1, x_2, \cdots$ denotes a sequence of feature vectors. There are other variants of the dynamic feature, such as fitting a linear regression over a window covering several frames. It is obvious that they fit in the linear transformation category in the same way.

It's striking that linear transformation plays such a central role in the feature extraction process. As any combination of linear operations is still linear, one starts to wonder whether the traditional front end can be simplified. Combining two linear transforms per se is no problem since a matrix multiplication is all we need. However, if we want to eliminate, say, DCT entirely, (i.e. from both training and decoding), we want to ensure that the performance won't be affected negatively. At a first glance, this seems to be the case: since LDA/MLLT is optimized for its own objective, any linear transformation before them should not make any difference to the final feature vector.

In the next section, we will formally analyze the invariance property of recognition performance with respect to linear transformations in the feature space. The simplified front end is presented in Section 3. In Section 4, we try to optimize the Mel-scale filtering step — another LT in the front end. Experiment results are presented in Section 5.

2. INVARIANCE PROPERTIES OF SPEECH RECOGNIZERS

Under certain modeling schemes, recognition performance is insensitive to linear transformations in the feature space. Two important notes:

- The boundary between acoustic model and front end is not clear cut. From now on, we will consider LDA and/or MLLT part of the acoustic modeling scheme, rather than LTs in the feature space, unless stated otherwise.
- When comparing two different front ends, retraining is a must. Thus *invariance* hereafter always means invariance after retraining.

For a simple example, consider diagonal (*not* diagonalization!) LT, i.e. stretching/compressing each dimension independently. Although the measured likelihood will change for sure, recognition performance won't be affected under most modeling schemes. In fact, that is why CMN schemes need not worry about whether to normalize the variance to 1 or 0.5, so long as it is consistently normalized. But for general LTs, a detailed discussion is in order, depending on the particular acoustic modeling scheme:

1. Plain Diagonal-Covariance: invariant only to diagonal LT

   Diagonal covariance assumes independence among feature dimensions. Any scaling of a certain dimension will be absorbed as scaling of the corresponding variance parameter of the Gaussian:

   $$X \sim N(\mu, \sigma^2) \implies aX \sim N(a\mu, (a\sigma)^2)$$

   where X is a random variable, a is the scaling factor.

   However, any transformation beyond scaling, such as rotation and affine transformation, may result in a performance difference.

2. Full-Covariance Gaussian: invariant to all LTs

   If the ML estimate for feature vector $x, x \in R^d$ is (μ, Σ), after a linear transformation $y = Ax$, the ML estimate for y becomes $(A\mu, A\Sigma A^T)$. For any test vector x,

   $$p(x|\mu, \Sigma) = |A| * p(Ax|A\mu, A\Sigma A^T)$$

   It is clear that the likelihood is scaled by a constant $|A|$. This will not affect discrimination among models. Therefore recognition performance won't be affected, although some hard-wired decoder parameters may need to be adjusted (beam size, language model weight, etc.)

3. Semi-Tied Covariance / MLLT: invariant to all LTs

   Note a full covariance matrix can be decomposed as a diagonal covariance matrix plus a rotation

   $$\Sigma = U\Sigma_d U^T$$

   where Σ_d is the diagonal term and U is a rotation ($UU^T = I$).

   By assigning each model a diagonal term and tying the rotation matrix among models (which may no longer be a rotation), we get the semi-tied covariance [2], and/or heteroscedastic LDA [4]. All parameters are estimated in the ML fashion [3].

   As in the full covariance case, linear transformation in the feature space $y = Ax$ results in

   $$\mu_y = A\mu \quad \Sigma_y = A\Sigma A^T = AU\Sigma_d U^T A^T$$

and

$$p(x|\mu, \Sigma_d, U) = |A| * p(Ax|A\mu, \Sigma_d, AU)$$

Thus it is also invariant to LT.

4. Diagonal Covariance + LDA: invariant to all LTs?

   LDA was initially introduced as a dimensional reduction technique that tries to retain most of the discrimination power in a reduced space.

   The criterion for choosing the LDA matrix is

   $$\arg\max_B \frac{|B\Sigma_b B^T|}{|B\Sigma_w B^T|}$$

   where Σ_b is the between-class scatter matrix, Σ_w is the within-class scatter matrix.

   Note the LDA solution is not unique, i.e. if B is found to maximize the criterion, AB is also a solution, so long as A is non-singular. In other words, LDA only defines the optimal subspace, regardless of any transformation within that subspace. This is exactly why it helps to have an additional diagonalization transform on top of LDA. However, given a particular LDA implementation[2] (which usually returns a single solution), we might expect that the uniqueness of the transform is guaranteed empirically.

   Under this assumption, it's easy to see that any non-singular feature space LT A (before LDA) will be absorbed into the new LDA matrix, yielding exactly the same feature (and therefore the same model parameters):

   $$\begin{aligned} B' &= \arg\max_B \frac{|BA\Sigma_b A^T B^T|}{|BA\Sigma_w A^T B^T|} \\ \Longrightarrow \quad B'A &= \arg\max_B \frac{|B\Sigma_b B^T|}{|B\Sigma_w B^T|} \end{aligned}$$

3. SIMPLIFYING THE FRONT END

When a chosen modeling scheme is invariant to LTs in the feature space, we can eliminate unnecessary LTs in the front end without any loss in recognition accuracy.

It's easy to see that in the traditional front end, everything after the log can be consolidated into one single LT (on the extended vector which is a concatenation of several adjacent frames). Since CMN is not a global LT but rather cluster-dependent, it can't be absorbed into the single LT described above. However, it can be shown that in a series of LTs, it doesn't matter where exactly mean subtraction is done.[3]

Taking the front end in Figure 1, we can see the net effect of CMS (Cepstral Mean Subtraction) is to make the mean of feature vectors equal 0. We can just as well move the mean subtraction step one step before or after: immediately after the log, or after LDA. The resulted feature vector will stay the same.

This property, together with invariance properties established above, allows us to consolidate all the LT operations into a single LT plus mean subtraction, illustrated below:

FFT → Mel-scale filterbank → log → CMS → LDA/MLLT

In this simplified front end, we don't even need the concept of DCT, nor that of cepstrum. Of course, if the modeling scheme is diagonal covariance without LDA/MLLT, the DCT step still makes a difference as it compensates to a certain extent for the dimensional independence assumption.

4. OPTIMIZING THE FRONT END

The previous section concludes that all the LTs after the log can be safely consolidated into a single LT. Now we will go one step further: optimize the front end. As stated above, mel-scale filterbank is just another LT. Thus the generalized front end looks like Figure 2. One would naturally start to question the optimality of mel-scale. After all, it's motivated perceptually, and is not necessarily consistent with the overall statistical framework.

Due to the nonlinearity of the log step, it's not straightforward how to optimize this stage (LT A) directly. Instead, we tried to leave out this stage completely, since the function it serves, namely smoothing the spectra and reducing dimensionality, can well be captured in LT C after the log. And better yet, that LT lends itself to easy optimization in a data-driven fashion. This leads to a greatly simplified and unconventional front end: the LLT front end (Figure 3).

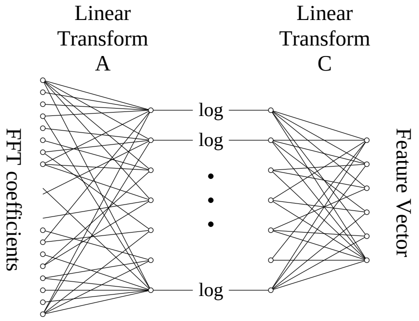

Figure 2: Generalized Front End

Another way to interpret the LLT front end is that LT A is delayed until after the log, and integrated into LT C. Note that there are actually more parameters in the system than before, due to the fact that the LT has to operate on the raw FFT spectrum rather than its reduced representation.

[2] The LDA algorithm implemented in JRTk is simultaneous diagonalization.

[3] Variance normalization can not be streamlined in the same way as mean subtraction. However, this may not be a major problem.

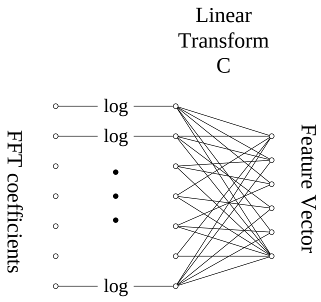

Figure 3: The LLT Front End

5. EXPERIMENTS

Experiments are carried out on the 1998 Hub4e evaluation set 1. The baseline recognizer is a quinphone system with 6000 distributions sharing 2000 codebooks, trained using JRTk[1]. There are about 105k Gaussians in the system. VTLN, cluster-based CMN, and a 7-frame context window are used. LDA is applied to reduce feature dimensionality to 42, followed by an optional diagonalization transform (MLLT). All results are first-pass decoding results using a 40k lexicon and a trigram language model.

Table 1 compares the standard MFCC front end (Figure 1) with the one without DCT. The other subtle difference between the two is that variance normalization is done for the former but not the latter. In both cases we used LDA, but no MLLT.

In Table 2 we compared the LLT front end (Figure 3) with the MFCC front end. LDA+MLLT are used in both cases. We see a 5% relative improvement by using the LLT front end.

System	Avg	F0	F1	F2	F3	F4	FX
MFCC	21.6	10.2	21.8	31.2	34.3	16.5	31.7
w/o DCT	21.6	10.2	20.7	30.8	36.6	16.3	32.6

Table 1: WER(%) on Hub4e98 Set1 (both without MLLT)

System	Avg	F0	F1	F2	F3	F4	FX
MFCC	20.0	9.4	21.1	32.6	30.2	15.1	28.3
LLT	19.0	9.2	20.0	29.2	27.5	14.2	27.2

Table 2: WER(%) on Hub4e98 Set1 (both with MLLT)

The improvement should be interpreted as evidence of the advantage of data-driven methods over their ad hoc counterparts. Although DCT, Mel-scale filtering are eliminated from explicit calculation, we believe their effects are well captured (and even optimized) by the LT trained from data. For DCT, which is mainly used to decorrelate among feature dimensions, the diagonalization transform is surely doing a much better job; for the Mel-scale filterbank, whose effect is to smooth the spectrum, we believe it's delayed and integrated into the single linear transform after log.

Also, the reason behind having both LDA and MLLT in the training process is that we don't have a feasible solution on how to jointly reduce dimensionality and maximize likelihood simultaneously.

6. RELATED WORK & CONCLUSION

Like many others, we believe that front-end parameterization and acoustic modeling should be considered jointly. In fact, the necessity of the DCT step has been questioned by many researchers, for example, [5].

This paper first gave a novel view of the front end in terms of linear transformation. Then we formally proved the invariance property and experimentally verified that the DCT step can be omitted. Last, we proposed a greatly simplified front-end scheme to optimize the Mel-scale filterbank.

There is still a lot to do towards optimizing the front end. For example, the coexistence of LDA and MLLT, each optimized for a different criterion, definitely calls for a better integration. It has been shown that different criteria for the dimensional reduction stage can lead to better performance [5].

7. ACKNOWLEDGEMENTS

Thanks to Hagen Soltau and Jürgen Fritsch for helping with the semi-tied covariance code, and many others for helpful discussions.

8. REFERENCES

[1] M. Finke, J. Fritsch, P. Geutner, K. Ries, and T. Zeppenfeld. The JanusRTk Switchboard/Callhome 1997 evaluation system. In *Proc. of LVCSR Hub5e Workshop*, 1997.

[2] M. J. F. Gales. Semi-tied covariance matrices. In *Proc. ICASSP98*, 1998.

[3] R. Gopinath. Maximum likelihood modeling with gaussian distributions for classification. In *Proc. ICASSP98*, 1998.

[4] N. Kumar. *Investigation of Silicon Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*. PhD thesis, Johns Hopkins University, 1997.

[5] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen. Maximum likelihood discriminant feature spaces. In *Proc. ICASSP2000*, 2000.

[6] S. Wegmann, P. Zhan, and L. Gillick. Progress in broadcast news transcription at dragon systems. In *Proc. ICASSP99*, 1999.