NOTES FOR AFFINE TRANSFORM-BASED VTLN

Daniel Povey

Microsoft,
One Microsoft Way, Redmond, WA 98052
dpovey@microsoft.com
(Written in 2010)

ABSTRACT

These are some notes on a linear transform based approximation to VTLN. It describes how, given a small amount of data for which we have the original and "conventionally warped" features, we can obtain a transform that approximates the original warping and leaves the mean and covariance of the sample data unchanged.

Index Terms— VTLN

1. INTRODUCTION

The idea is to approximate, as closely as possible, VTLN done in the conventional way, by an affine transform of the cepstra. This is a trained approximation based on data (i.e. a few utterances' worth of feature data). In addition we constrain the mean and covariance matrix of the transformed features to be the same as the un-transformed features (for the sampled data). This ensures that the warped features will be well-matched to the unwarped features.

The inputs to this process are as follows. We have the un-warped features $\mathbf{x}_t$, for $t = 1 \dots T$. We have N VTLN warp factors, typically 21 of these. These are used to generate VTLN-warped features $\mathbf{y}_t^{(n)}$ for $1 \leq n \leq N$, using a conventional feature-level computation (e.g., one based on moving the center positions of the mel-frequency filter banks). We use $\mathbf{x}^+$ to represent $\mathbf{x}$ with a 1 appended. We are training N affine transforms $\mathbf{T}^{(n)}$ that do the transform

$$\mathbf{x} \rightarrow \mathbf{T}^{(n)} \mathbf{x}^+. \tag{1}$$

We define $\mathbf{z}_t^{(n)}$ to be $\mathbf{T}^{(n)} \mathbf{x}_t$; this is the affine approximation to $\mathbf{y}_t^{(n)}$, and the idea is that we choose $\mathbf{T}^{(n)}$ to make $\mathbf{z}_t^{(n)} \simeq \mathbf{y}_t^{(n)}$, subject to mean and covariance constraints.

The reason for doing the affine approximation is partly for convenience, so we can store the un-warped features and use a simple transform to create the warped ones. It is also a convenient way to ensure that the mean and covariance of the features is the same across the warp factors, which helps to ensure that they can all be modeled reasonably well with the same model. We note that the motivation for keeping the mean and variance of the sample data unchanged after the warping is questionable. The reason is that we are going to apply the different warping functions to different sets of data. It makes sense that after the warping, we would want map all the original data subsets to have the same mean and covariance. But it does not follow from this, that we want a given subset of data to have the same mean and covariance after being warped by each of the individual warping functions. With this in mind, we explain further why we do it this way. The initial motivation is that the linear transform is, in practice, not a very good approximation of the conventionally applied warp (perhaps 20%-40% of the variance of the conventionally warped cepstra is left over after linear prediction, for the highest order cepstra and the more extreme warp factors). Therefore, we are concerned that the linear versions of the warped cepstra will have a lower overall variance than the unwarped ones and this will introduce various mismatches and biases. This is the initial motivation; we originally normalized the variance in each dimension, andwhat we describe here is a full-covariance extension of that, that normalizes the entire covariance matrix. Another motivation is as follows: when we were normalizing only the diagonal of the covariance, the resulting matrices had non-unit determinants: the log-determinants were zero for the central warped factor, going negative (e.g. around -0.5) for the most extreme warp factors (0.9 or 1.1). If we used these determinants as we "should" do, at the stage when we pick which warping matrix to use for a given utterance, we got poor performance VTLN and we noticed that the VLTN warp factors were clustered to closely around 1.0. It was better to ignore the log-determinant. If we compute the linear approximations the way we describe here, the log-determinant is zero and hence there is no need to ignore it. Also we were concerned that without correcting for the full covariance, the warping with more extreme warp factors might produce variances that were too low in some off-diagonal directions, and when we use more advanced models with full covariances, these might be hard to model with a single covariance matrix across all warp factors and would result in extra parameters being allocated by the model simply to account for the different warping functions.

We feel we should mention at this point that the "right" way to solve this problem, from our point of view, is: after determining the mapping from utterances to PVTLN warp factors (or interleaved with determining this mapping), use a Maximum Likelihood solution based on fMLLR to compute the best transform for the class. Unfortunately our initial investigations in this direction were not very promising.

2. COMPUTING THE TRANSFORM MATRIX: DERIVATION

Now we describe the way we compute the transformation matrix $\mathbf{T}^{(n)}$ (for the warping class index n). The inputs to the process are $\mathbf{x}_t$ and $\mathbf{y}_t^{(n)}$ for $1 \leq t \leq T$, and the output is the transformation matrix $\mathbf{T}^{(n)}$. At this point we drop the superscript (n) and make it implicit.

The formulation is as follows: we want to compute $\mathbf{T}^{(n)}$ to minimize an appropriately weighted sum-squared distance between $\mathbf{z}_t$ and $\mathbf{y}_t$, subject to the constraint that $\mathbf{z}$ has the same mean and co-

variance as $\mathbf{x}$. We can write this as follows: subject to the constraints

$$\sum_{t=1}^{T} \mathbf{x}_t = \sum_{t=1}^{T} \mathbf{z}_t \quad (2)$$

$$\sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t^T = \sum_{t=1}^{T} \mathbf{z}_t \mathbf{z}_t^T, \quad (3)$$

minimize

$$\sum_{t=1}^{T} (\mathbf{z}_t - \mathbf{y}_t)^T \mathbf{K} (\mathbf{z}_t - \mathbf{y}_t) \quad (4)$$

where $\mathbf{K}$ is an appropriate kernel matrix. We set $\mathbf{K}$ to the inverse covariance matrix of $\mathbf{x}_t$ (which is the same as the inverse covariance matrix of $\mathbf{z}_t$). This ensures that the resulting inner product is unaffected by affine transforms of the original features, which is a desirable kind of invariance.

Let us write

$$\mathbf{T} = [\mathbf{M}; \mathbf{v}], \quad (5)$$

so

$$\mathbf{z}_t = \mathbf{T}\mathbf{x}_t^+ = \mathbf{M}\mathbf{x}_t + \mathbf{v}. \quad (6)$$

Let $\bar{\mathbf{x}}$ be the mean of $\mathbf{x}$, i.e.

$$\bar{\mathbf{x}} = \frac{1}{T} \sum_t \mathbf{x}_t. \quad (7)$$

Let us define

$$\hat{\mathbf{x}}_t = \mathbf{x}_t - \bar{\mathbf{x}} \quad (8)$$

$$\hat{\mathbf{z}}_t = \mathbf{z}_t - \bar{\mathbf{x}} \quad (9)$$

$$\hat{\mathbf{y}}_t = \mathbf{y}_t - \bar{\mathbf{x}}. \quad (10)$$

We can rewrite Equations (2), (3) and (4) with $\hat{\mathbf{x}}_t$ in place of $\mathbf{x}_t$ and the same for $\mathbf{y}$ and $\mathbf{z}$, and they are equivalent to the original equations (i.e. the constraints are true for for the same values of $\mathbf{T}$ and Equation 4 will have the same value when rewritten. Because of the mean constraint, the mean of $\hat{\mathbf{z}}$ must equal zero. Written as an affine function of $\hat{\mathbf{x}}$, the offset term would be zero: therefore we can make $\hat{\mathbf{z}}$ a linear (not affine) function of $\hat{\mathbf{x}}$, and write:

$$\hat{\mathbf{z}}_t = \mathbf{M}\hat{\mathbf{x}}_t. \quad (11)$$

Let us write the covariance of $\hat{\mathbf{x}}$ as:

$$\mathbf{S} = \frac{1}{T} \sum_{t=1}^{T} \hat{\mathbf{x}}\hat{\mathbf{x}}^T, \quad (12)$$

recalling the $\hat{\mathbf{x}}$ has zero mean. Consider the constraints (2) and (3). The first constraint is automatically satisified because $\hat{\mathbf{z}} = \mathbf{M}\hat{\mathbf{x}}$ ensures that $\hat{\mathbf{z}}$ is zero-mean. The second constraint becomes:

$$\frac{1}{T} \sum_{t=1}^{T} \hat{\mathbf{z}}_t \hat{\mathbf{z}}_t^T = \mathbf{S}. \quad (13)$$

It is helpful to perform a further change of variables so that the unit matrix appears on the right of (13). Let us do the Cholesky decomposition:

$$\mathbf{S} = \mathbf{C}\mathbf{C}^T, \quad (14)$$

where the lower triangular matrix $\mathbf{C}$ is the Cholesky factor of $\mathbf{S}$. Define

$$\tilde{\mathbf{x}}_t = \mathbf{C}^{-1}\hat{\mathbf{x}}_t \quad (15)$$

$$\tilde{\mathbf{y}}_t = \mathbf{C}^{-1}\hat{\mathbf{y}}_t \quad (16)$$

$$\tilde{\mathbf{z}}_t = \mathbf{C}^{-1}\hat{\mathbf{z}}_t. \quad (17)$$

The relationship between $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{z}}_t$ is given by substituting $\hat{\mathbf{x}} = \mathbf{C}\tilde{\mathbf{x}}_t$ and $\hat{\mathbf{z}} = \mathbf{C}\tilde{\mathbf{z}}_t$ into (11) and then left-multiplying by $\mathbf{C}^{-1}$: we get

$$\tilde{\mathbf{z}}_t = \mathbf{C}^{-1}\mathbf{M}\mathbf{C}\tilde{\mathbf{x}}_t \quad (18)$$

$$= \mathbf{N}\tilde{\mathbf{x}}_t, \quad (19)$$

$$\mathbf{N} \equiv \mathbf{C}^{-1}\mathbf{M}\mathbf{C}. \quad (20)$$

Multiplying Equations (12) and (13) on the left by $\mathbf{C}^{-1}$ and on the right by $\mathbf{C}^{-T}$, we have:

$$\frac{1}{T} \sum_{t=1}^{T} \tilde{\mathbf{x}}_t \tilde{\mathbf{x}}_t^T = \mathbf{I} \quad (21)$$

$$\frac{1}{T} \sum_{t=1}^{T} \tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^T = \mathbf{I}. \quad (22)$$

Using $\tilde{\mathbf{z}}_t = \mathbf{N}\tilde{\mathbf{x}}_t$, it follows from Equations (21) and (22) that

$$\mathbf{N}\mathbf{N}^T = \mathbf{I}, \quad (23)$$

i.e. $\mathbf{N}$ is an orthogonal matrix. We have thus simplified the variance constraint to an orthogonality condition on the transform $\mathbf{N}$. Next we turn to the objective function we are minimizing. When we write Equation (4) in the new variables, it becomes:

$$\sum_{t=1}^{T} (\tilde{\mathbf{z}}_t - \tilde{\mathbf{y}}_t)^T (\tilde{\mathbf{z}}_t - \tilde{\mathbf{y}}_t). \quad (24)$$

Here, the kernel matrix $\mathbf{K} = \mathbf{S}^{-1}$ cancels because when we use $\hat{\mathbf{z}}_t = \mathbf{C}\tilde{\mathbf{z}}_t$ and the same for $\hat{\mathbf{y}}_t$, we get a middle factor $\mathbf{C}^T\mathbf{S}^{-1}\mathbf{C} = \mathbf{I}$. Equation (24) can be written as three terms: two squared terms and a cross term. The squared terms are irrelevant to the optimization because $\sum_{t=1}^{T} \tilde{\mathbf{z}}_t^T \tilde{\mathbf{z}} = T\,\mathrm{tr}\,(\mathbf{I})$ which is a constant, and $\sum_{t=1}^{T} \tilde{\mathbf{y}}_t^T \tilde{\mathbf{y}}$ is a data-dependent quantity that is independent of $\mathbf{N}$. This leaves us with the following objective to be minimized:

$$-2 \sum_{t=1}^{T} \tilde{\mathbf{y}}_t^T \tilde{\mathbf{z}}_t. \quad (25)$$

This is equivalent to maximizing:

$$\sum_{t=1}^{T} \tilde{\mathbf{y}}_t^T \tilde{\mathbf{z}}_t \quad (26)$$

$$= \sum_{t=1}^{T} \tilde{\mathbf{y}}_t^T \mathbf{N} \tilde{\mathbf{x}}_t \quad (27)$$

$$= \mathrm{tr}\left(\mathbf{N}\left(\sum_{t=1}^{T} \tilde{\mathbf{x}}_t \tilde{\mathbf{y}}_t^T\right)\right) \quad (28)$$

$$= \mathrm{tr}\,(\mathbf{N}\mathbf{P}), \quad (29)$$

$$\mathbf{P} \equiv \sum_{t=1}^{T} \tilde{\mathbf{x}}_t \tilde{\mathbf{y}}_t^T. \quad (30)$$

The quantity $\mathbf{P}$ is accumulated from the data, and will determine the optimal value of $\mathbf{N}$. We do the singular value decomposition on $\mathbf{P}$:

$$\mathbf{P} = \mathbf{U}\mathbf{L}\mathbf{V}^T, \quad (31)$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal and $\mathbf{L}$ is nonnegative and diagonal. Let us define:

$$\mathbf{Q} \equiv \mathbf{V}^T\mathbf{N}\mathbf{U}. \quad (32)$$

This implies:

$$\mathbf{N} = \mathbf{V}\mathbf{Q}\mathbf{U}^T, \tag{33}$$

which we get multiplying (32) on the left by $\mathbf{V}$ and the right by $\mathbf{U}^T$ and using orthogonality of $\mathbf{U}$ and $\mathbf{V}$ to cancel. Thus, we can write function we are maximizing, $\mathrm{tr}\,(\mathbf{NP})$, as:

$$\mathrm{tr}\,(\mathbf{V}\mathbf{Q}\mathbf{U}^T\mathbf{U}\mathbf{L}\mathbf{V}^T) = \mathrm{tr}\,(\mathbf{Q}\mathbf{L}\mathbf{V}^T\mathbf{V}) = \mathrm{tr}\,(\mathbf{Q}\mathbf{L}). \tag{34}$$

Here, $\mathbf{L}$ is a known diagonal matrix that arises from the SVD of $\mathbf{P}$, and $\mathbf{Q}$ is a matrix that we are solving for (that determines $\mathbf{N}$). It follows from the orthogonality of $\mathbf{N}$, $\mathbf{U}$ and $\mathbf{V}$, and Equation 32, that $\mathbf{Q}$ is orthogonal (e.g. write out $\mathbf{Q}\mathbf{Q}^T$ substituting in (32) and all factors cancel). Because $\mathbf{L}$ is nonnegative and diagonal, $\mathrm{tr}\,(\mathbf{QL})$ is maximized by having the diagonal entries of $\mathbf{Q}$ as positive as possible. These diagonal entries cannot be more than 1 (by orthogonality), so the objective function is maximized by having them all equal 1. This implies that the off-diagonal terms are zero, so $\mathrm{tr}\,(\mathbf{QL})$ is maximized by $\mathbf{Q} = \mathbf{I}$.

The rest of the work consists of substituting back to get the original transform. Using $\mathbf{Q} = \mathbf{I}$, Equation (33) becomes:

$$\mathbf{N} = \mathbf{V}\mathbf{U}^T. \tag{35}$$

Multiplying (20) on the left by $\mathbf{C}$ and the right by $\mathbf{C}^{-1}$, we have:

$$\mathbf{M} = \mathbf{C}\mathbf{N}\mathbf{C}^{-1}. \tag{36}$$

Next we show how to obtain $\mathbf{T}$ from $\mathbf{M}$. $\mathbf{T}$ operates on the original features $\mathbf{x}$ which differ from $\hat{\mathbf{x}}$ by a mean offset. Substituting (8) and (9) into $\hat{\mathbf{z}} = \mathbf{M}\hat{\mathbf{x}}$,

$$\begin{aligned}
\mathbf{z}_t - \bar{\mathbf{x}} &= \mathbf{M}\,(\mathbf{x}_t - \bar{\mathbf{x}}) && (37)\\
\mathbf{z}_t &= \mathbf{M}\mathbf{x}_t - \mathbf{M}\bar{\mathbf{x}} + \bar{\mathbf{x}} && (38)\\
\mathbf{z}_t &= \mathbf{T}\mathbf{x}_t^+, && (39)\\
\mathbf{T} &\equiv [\mathbf{M}\,;\,\mathbf{v}] && (40)\\
\mathbf{v} &\equiv \bar{\mathbf{x}} - \mathbf{M}\bar{\mathbf{x}}. && (41)
\end{aligned}$$

3. COMPUTING THE TRANSFORM MATRIX: SUMMARY

Here we summarize the computation. The input to the process is a sequence of un-transformed features $\mathbf{x}_t$ for $1 \le t \le T$, and for each warping class n, a sequence of warped features $\mathbf{y}_t^{(n)}$. We aim to compute a transform $\mathbf{T}^{(n)}$ that takes $\mathbf{x}_t^+$ to $\mathbf{y}_t^{(n)}$ as closely as possible while leaving the mean and covariance of $\mathbf{x}$ unchanged.

Firstly, we compute statistics of $\mathbf{x}$:

$$\begin{aligned}
\bar{\mathbf{x}} &:= \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t && (42)\\
\mathbf{S} &:= \left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t\mathbf{x}_t^T\right) - \bar{\mathbf{x}}\bar{\mathbf{x}}^T && (43)
\end{aligned}$$

We compute the Cholesky factor $\mathbf{C}$ such that:

$$\mathbf{S} = \mathbf{C}\mathbf{C}^T. \tag{44}$$

For each n, the computation is as follows (and we are dropping the superscript (n) here that appears on all quantities except $\mathbf{C}$, $\mathbf{x}_t$ and $\bar{\mathbf{x}}$):

$$\mathbf{P}_0 := \left(\sum_{t=1}^{T}\mathbf{x}_t\mathbf{y}_t^T\right) - \bar{\mathbf{x}}\sum_{t=1}^{T}\mathbf{y}_t^T, \tag{45}$$

so that $\mathbf{P}_0 = \sum_{t=1}^{T}\hat{\mathbf{x}}_t\mathbf{y}_t^T = \sum_t \hat{\mathbf{x}}_t\hat{\mathbf{y}}_t^T$, where the second equality uses the fact that the mean of $\hat{\mathbf{x}}_t$ is zero. Then we compute

$$\mathbf{P} := \mathbf{C}^{-1}\mathbf{P}_0\mathbf{C}^{-T}. \tag{46}$$

We do the singular value decomposition:

$$\mathbf{P} = \mathbf{U}\mathbf{L}\mathbf{V}^T. \tag{47}$$

Here, the diagonal elements of $\mathbf{L}$ are a useful diagnostic; if they are close to T, then the variance constraint is not affecting the sum-of-squares objective function very much (versus having no variance constraint). We compute:

$$\begin{aligned}
\mathbf{N} &:= \mathbf{V}\mathbf{U}^T && (48)\\
\mathbf{M} &:= \mathbf{C}\mathbf{N}\mathbf{C}^{-1} && (49)\\
\mathbf{v} &:= \bar{\mathbf{x}} - \mathbf{M}\bar{\mathbf{x}} && (50)\\
\mathbf{T} &:= [\mathbf{M}\,;\,\mathbf{v}]\,. && (51)
\end{aligned}$$