LU FACTORIZATION FOR FEATURE TRANSFORMATION

Patrick Nguyen[1,2], *Luca Rigazio*[1], *Christian Wellekens*[2] *and Jean-Claude Junqua*[1]

[1] Panasonic Speech Technology Laboratory
Santa Barbara, U.S.A.
{nguyen, rigazio, jcj}@research.panasonic.com

[2] Institut Eurécom
Sophia-Antipolis, France
welleken@eurecom.fr

ABSTRACT

Linear feature space transformations are often used for speaker or environment adaptation. Usually, numerical methods are sought to obtain solutions.

In this paper, we derive a closed-form solution to ML estimation of full feature transformations. Closed-form solutions are desirable because the problem is quadratic and thus blind numerical analysis may be lured into the recess of numerically valid, albeit poor, attractors. We decompose the transformation into upper and lower triangular matrices, which are estimated alternatively using the EM algorithm. Furthermore, we extend the theory to Bayesian adaptation.

On the Switchboard task, we obtain 1.6% WER improvement by combining the method with MLLR, or 4% absolute using adaptation.

1. INTRODUCTION

Linear feature space transformations have been subject to intense investigation recently. They provide a conceptually appropriate way of normalizing environment or speaker mismatch. They are naturally integrated into the SAT paradigm towards offering compact models for speech recognition.

The analytical mathematics are somewhat related to semitied covariances [1] and MLLT [2]. Both acknowledge the absence of a closed-form solution in the general case and proceed to define numerical expedient for that ailment. Numerical methods are sensitive to conditioning and extra care is admonished to ensure convergence. Additionally, more insight may be gained from analytic solutions.

In this paper, we discover a non-trivial special case of linear transformations that admits a closed-form solution: triangular matrices. We generalize to a full matrix by alternating the estimation of upper and lower triangular matrix, in a pattern which mimics the LU factorization.

Lastly, we define the MAP estimator which serves as a foundation for smoothing.

2. FEATURE-SPACE TRANSFORMATIONS

2.1. Linear transformation of observations

Let X be a random variable with pdf $p_X(\cdot)$. A corollary of the plug-in rule for pdfs yields:

$$Y = AX + b \Rightarrow p_Y(y) = |A| p_X(y). \tag{1}$$

Note that the bias b does not appear in the jacobian. Therefore, model-space offset is equivalent to (opposite sign) feature-space offset. This is a well-known area and we will discard the bias in most derivations for simplicity. The plug-in rule may be stated as: plug in the transformed observation in the pdf and divide by the Jacobian $|A|^{-1}$.

2.2. In the EM algorithm

The mathematics of Hidden Markov Models (HMMs) are well-known. Using the plug-in rule, we re-compute the expected log-likelihood Q. The Q function becomes

$$Q = -\frac{1}{2} \sum_{t,m} \gamma_m(t) \left\{ -\log |A|^2 + (\mu - Ao_t)^T R (\mu - Ao_t) \right\}$$

and its derivative

$$\frac{\partial Q}{\partial A} = -\sum_{t,m} \gamma_m(t) \left\{ -A^{-T} + RAo_t o_t^T - R\mu o_t^T \right\}. \tag{2}$$

We know that stationary points of the gradient correspond to a maximum or minimum in Q. This seemingly simple problem is a multidimensional quadratic equation and has no closed-form solution in general. Gales [3] assumes rows to be almost independent and optimizes row by row. Gopinath [2] points out that half of the function is quadratic and therefore suitable for conjugate gradient descent. Digilakis [4] advocates iterative numerical methods but cites none in particular.

2.3. Diagonal matrix

When the matrix A is diagonal, then there are two solutions per dimension. We also assume precision matrices R to be diagonal. Let a_{dd} be the d^{th} diagonal element of A. The expression for the gradient is quadratic and may be found in Gales or

Digilakis. However, neither of them seem to give an explicit expression nor bear a preference for either root. We choose:

$$a_{dd} = \frac{1}{2}\left(\sqrt{\beta^2 + 4\eta} + \beta\right) \tag{3}$$

with the appropriate definitions of β and η:

$$\alpha = \sum_{t,m} \gamma_m(t) r_d o_d^2,$$
$$\beta = \alpha^{-1} \sum_{t,m} \gamma_m(t) r_d o_d \mu_d,$$
$$\eta = \alpha^{-1} \sum_{t,m} \gamma_m(t).$$

The Hessian (second derivative) in this case indicates which of the two solutions corresponds to a stable point by indicating more negative values:

$$\frac{\partial^2 Q}{(\partial a_{dd})^2} = -\sum_{t,m} \gamma_m(t) \left\{ \frac{1}{a^2} + r_d o_d^2 \right\} < 0. \tag{4}$$

Both roots of the characteristic equation correspond to maxima in the likelihood. However, our choice guarantees a smaller absolute value of the second derivative, and also a value closer to unity. Without this additional hint, numerical methods would converge arbitrarily to one of 2^N stationary points. The closed-form solution affords more insight.

2.4. Upper-triangular matrix and its closed-form solution

Since all rows of the matrix are independent, thanks to the diagonality of covariances, we may set a dimension d and solve each dimension independently. Let $a_k, k = d, d+1, ...N$ be the non-zero elements of the d^{th} row of A. Let b be the bias of the feature d. Define

$$a^* = [a_{d+1}, a_{d+2}, ..., a_N, b]^T, \tag{5}$$
$$o^* = [o_{d+1}, o_{d+2}, ..., o_N, 1]^T. \tag{6}$$

We seek to find $[a_d, a^*]$. Since the determinant only depends on a_d, it is treated differently. First, we solve a $(N-d) \times (N-d)$ linear subsystem for a^* using the $N-d$ last elements of the gradient. Then, we use the special equation for a_d to yield the quadratic form of the previous section.

The objective function for the dimension d is

$$Q = -\frac{1}{2}\left\{ -\log|a_d|^2 + (a^{*T} o^* + a_d o_d - \mu_d)^2 r_d \right\} \tag{7}$$

Differentiating with respect to a_k, $k = d+1, ..., N$ and b, we get a linear system

$$\frac{\partial Q}{\partial a^*} = -\sum_{t,m} \gamma_m(t) r_d \left[\mu_d - a^{*T} o^* - a_d o_d \right] o^*. \tag{8}$$

It is solved by:

$$a^* = M^{-1}(a_d y + z), \tag{9}$$

with the following

$$M = \sum_{t,m} \gamma_m(t) r_d o^* o^{*T} > 0,$$
$$y = \sum_{t,m} \gamma_m(t) r_d o_d \, o^*,$$
$$z = \sum_{t,m} \gamma_m(t) r_d \mu_d \, o^*.$$

Now we need to find a_d and substitute back.

The solution for a_d is found using the last derivative, which is merely a generalization of the diagonal case:

$$\frac{\partial Q}{\partial a_d} = \sum_{t,m} \gamma_m(t) r_d \Big[a_d^{-1} + (\mu_d - a^{*T} o^* - a_d o_d)(o_d + y^T M^{-1} o^*) \Big].$$

We can use the linear dependency specified by eq.(9), and finally state that a_d is again the solution of a quadratic expression

$$a_d = \frac{1}{2}\left(\beta + \sqrt{\beta^2 + 4\eta}\right), \tag{10}$$

with

$$\alpha = \sum_{t,m} \gamma_m(t) r_d o_d^2 - y^T M^{-1} y > 0,$$
$$\beta = \alpha^{-1}\left[\sum_{t,m} \gamma_m(t) r_d o_d \mu_d - y^T M^{-1} z \right],$$
$$\eta = \alpha^{-1} \sum_{t,m} \gamma_m(t) > 0.$$

When covariances are not diagonal, we must first solve the quadratic equation for a_{NN}. Then, knowledge of this coefficient will help find $a_{N-1,N-1}$ and $a_{N-1,N}$. We proceed thus upwards until the top row, in the same fashion as the back substitution step in a Gauss-Jordan matrix inversion.

2.5. The LU decomposition

Looking at eq.(2), we see that the crux of the problem resides in the presence of a log determinant, which implies in turn the presence of the inverse matrix. A common way of dealing with inverse matrices involves the LU decomposition of a matrix, that is to say, our matrix A is written as

$$A = LU \tag{11}$$

with U an upper-triangular matrix, and L a unitary, lower-triangular matrix. The diagonal elements of L are all equal to 1.

We embed this decomposition by alternating the maximization step in the EM algorithm:

$$o' = Ao = L(Uo). \tag{12}$$

The upper-triangular method was derived above, and the lower-triangular method is a special case of MLLR.

3. BAYESIAN EXTENSION

The Bayesian framework is useful for parameter smoothing. For instance, while using regression trees to define multiple classes, the leaf transforms are derived by smoothing with the parent nodes, as shown on Figure 1.

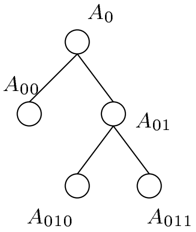

Fig. 1. Using a regression tree: transformations A_{00} and A_{01} are interpolated versions between ML and A_0.

The MAP framework is usually greatly simplified by selecting the prior distribution $p_0(A)$ among the family of conjugate priors for A.

MAP estimators and prior distributions were defined for all but the diagonal term. The conjugate prior for the bias is a Normal law. The conjugate prior for non-diagonal elements is elliptic. The probability of diagonal terms has a transcendent shape. The prior family does not appear frequently enough in nature to justify a name. We proceed to define it.

3.1. The Maxwell-Rayleigh-Normal distribution

A subset of the family of conjugate priors is a mixture of (extended) Maxwell, Rayleigh, and Gaussian distribution. We christen it hence the *Maxwell-Rayleigh-Normal* (MRN) distribution.

Maxwell's distribution models speeds of molecules in thermal equilibrium. It is defined for $x \geq 0, a > 0$:

$$p_M(x|a) = \sqrt{\frac{2}{\pi}} a^{3/2} x^2 e^{-ax^2/2}. \tag{13}$$

Furthermore, the Rayleigh distribution models the attenuation in fading channels and is

$$p_R(x|s) = \frac{x}{2s^2} e^{-\frac{1}{2}x^2/s^2}. \tag{14}$$

Lastly, the Normal distribution is an old acquaintance of ours

$$p_G(x|s) = \frac{1}{\sqrt{2\pi}s} e^{-\frac{1}{2}x^2/s^2}. \tag{15}$$

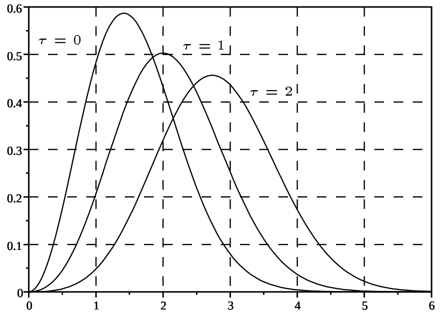

Fig. 2. The MRN law for different values of ν

We define the MRN distribution to be

$$p_{MRN}(x|\nu) = \Phi^{-1} x^2 e^{-(x-\nu)^2}. \tag{16}$$

The regularization constant Φ is chosen such that

$$\int dp_{MRN} = 1; \tag{17}$$

and we include it here for the sake of completeness

$$\Phi(\nu) = 2\sqrt{\pi} \left\{ 1 + \text{erf}(\frac{\nu}{2}) - \frac{1}{\sqrt{\pi}} e^{\nu^2/4} + \nu^2 \right\} +$$
$$\frac{1}{2}\nu \left\{ 1 - \frac{1}{8}\nu^2 e^{-\nu^2/4} \right\} + \frac{1}{2}\nu^2 \text{erf}(\nu/4).$$

and the error function $\text{erf}(x) = \int_0^x dy e^{-y^2}$. The distribution is shown on Figure 2. The value of the hyper-parameter ν with respect to the mean is shown on Figure 3.

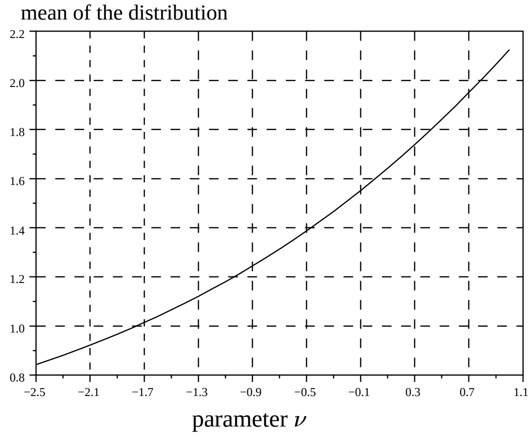

Fig. 3. The mean of MRN w.r.t. ν. The parameter that corresponds to identity is $\nu = -1.8$.

We proceed by defining the *raised MRN* law constitutes a family of conjugate priors:

$$p_{RMRN}(x|\nu, \tau) = \Phi_R^{-1}(\nu, \tau) x^{2\tau} e^{-\tau(x-\nu)^2}. \tag{18}$$

Unfortunately, unless τ is a multiple of $\frac{1}{2}$, moments have no

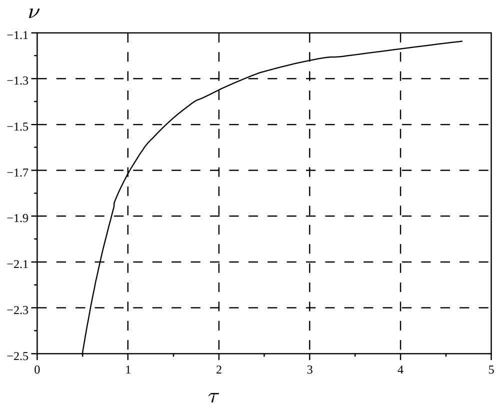

Fig. 4. We select τ, and choose ν s.t. the mean is one.

closed-form expression. Nevertheless, since we are only interested in values of τ, ν such that

$$\int dx \left[x p_{RMRN}(x) \right] = 1, \tag{19}$$

it is easier to use numerical integration and tabulate $\nu(\tau)$. We would then obtain the curve shown on Figure 4. The parameter τ is interpreted as the weight given to prior information.

4. EXPERIMENTS

4.1. Conditions

To validate our algorithm, we used the Switchboard conversational telephone speech database. We report results on the first evaluation test set of 2001 [5], which contains 20 conversations from the Switchboard-I database. The acoustic frontend uses 27 PLP coefficients (8 pole model plus energy, and their first and second derivatives), which were normalized using side-based cepstral mean subtraction (CMS) and variance normalization. We train a total of 256k Gaussians with diagonal covariances, pooled in 3600 mixtures using decision trees.

The language model (LM) for this task is a trigram model containing compound words and frequent abbreviations [6]. It was kindly provided to us by Andreas Stolcke of SRI. It contains 34k words, 5M bigrams, and 12M trigrams.

Our recognizer, called EWAVES [7], is a lexical-tree based, gender-independent, word-internal context-dependent, trigram Viterbi decoder with bigram LM lookahead. For adaptation, we use the transcription of the first pass. The second pass is identical to the first pass but runs on adapted features or with adapted models.

4.2. Results

Table 1 shows the results. The feature space transformation, or MLLU for (Maximum-likelihood LU transformation), yields an improvement comparable with MLLR when used in isolation.

There is a .2% WER improvement if we only use block-diagonal matrices. We have observed that MLLR behaves best with 7 regression classes (1 for silence, 4 for vowels, and 2 for consonants). In this case as well, constraining the transformation matrices to be block-diagonal, we get an improvement.

When we use MLLU as a feature normalization, followed by MLLR model adaptation, we obtain a 1.6% WER improvement over the baseline MLLR adapted models.

	WER
SI	34.6%
MLLR 1 global class	32.8%
MLLU 1 global class	32.8%
MLLU block-diag	32.6%
MLLR 7 classes + block	32.2%
MLLU + MLLR(7)	30.6%

Table 1. Results

5. DISCUSSION AND FUTURE WORK

In this paper, we have exposed a closed-form solution for the case of linear feature space triangular transformations. We extended the algorithm in the EM algorithm to yield the LU factorization of a full linear transformation. Furthermore, the Bayesian framework was also explored.

On Switchboard, our new algorithm, MLLU, yields a significant improvement over adapted models. Due to time constraints, we were not able to investigate multiple-class, Bayesian LU feature decomposition.

6. REFERENCES

[1] M. J. F. Gales, "Adapting Semi-Tied Full-Covariance Matrix HMMs," Tech. Rep., Cambridge University (CUED), 1997.

[2] R. A. Gopinath, "Maximum Likelihood Modeling with Gaussian Distributions for Classification," in *Proc. of ICASSP'98*, Seattle.

[3] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition," Tech. Rep., Cambridge University (CUED), May 1997.

[4] V. Digilakis, D. Ritchev, and L. Neumeyer, "Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures," *IEEE Trans. SAP*, vol. 3, pp. 129–136, 1995.

[5] A. Martin and M. Przybocki, "Analysis of results," in *2001 NIST LVCSR Workshop*, 2001.

[6] A. Stolcke, H. Bratt, J. Butzberger H. Franco, V. Ramana Rao Gadde M. Plauch, C. Richey, E. Shriberg K. Snmez, F. Weng, and J. Zheng, "The SRI March 2000 Hub-5 Conversational Speech Transcription System," in *Proc. of 2000 Speech Transcription Workshop*, 2000.

[7] P. Nguyen, L. Rigazio, and J.-C. Junqua, "EWAVES: an efficient decoding algorithm for lexical tree based speech recognition," in *Proc. of ICSLP*, Beijing, China, Oct. 2000, vol. 4, pp. 286–289.

[8] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaption of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.