LASSO Model Adaptation for Automatic Speech Recognition

Jinyu Li JINYLI@MICROSOFT.COM
Microsoft Corporation, Redmond, WA. 98052 USA

Ming Yuan MYUAN@ISYE.GATECH.EDU
Chin-Hui Lee CHL@ECE.GATECH.EDU
Georgia Institute of Technology, Atlanta, GA. 30332 USA

Abstract

Inspired by the success of least absolute shrinkage and selection operator (LASSO) in statistical learning, we propose an L_1 regularized maximum likelihood linear regression (MLLR) to estimate models with only a limited set of adaptation data to improve accuracy for automatic speech recognition, by regularizing the standard MLLR objective function with an L_1 constraint. The so-called LASSO MLLR is a natural solution to the data insufficiency problem because the L_1 constraint regularizes some parameters to exactly 0 and reduces the number of free parameters to estimate. Tested on the 5k-WSJ0 task, the proposed LASSO MLLR gives significant word error rate reduction from the errors obtained with the standard MLLR in an utterance-by-utterance unsupervised adaptation scenario.

1. Introduction

Parameter adaptation is one of the most efficient techniques to address the potential mismatches between the training and testing environments in automatic speech recognition (ASR). Although maximum likelihood linear regression (MLLR) [1] and maximum a posteriori (MAP) [2] adaptation are very popular, they cannot be directly applied to some challenging application scenarios, such as voice search or voice mail transcription [3], in which only one utterance can be available for self-adaptation. The limited amount of data may not be enough for reliably estimating even one transformation matrix. There are two popular solutions to this data insufficiency problem. One is to prepare a prior distribution for the transformation, and the MAP criterion is used for matrix parameter estimation (e.g., MAPLR [4]). The other solution is to employ eigen-family methods, such as eigen-voice [5], eigen-MLLR [6]. In both cases we need either pre-computed prior densities or a collection of basis vectors/matrices.

In statistical learning, least absolute shrinkage and selection operator (LASSO) [7] has been demonstrated as an effective method to handle the data sparsity problem. By adding an L_1 constraint to the original objective function, LASSO can be very effective in controlling over-fitting because it shrinks parameter values to zero and reduces the degree of freedom to estimate them.

As stated in [1], the linear regression is a special case of MLLR. Therefore, it is straightforward to borrow the success of LASSO in linear regression to address the data sparsity problem in MLLR in which adaptation is performed with a linear transformation. In this paper, we propose LASSO MLLR, an adaptation method with the L_1 constraint. We give the solution and implementation detail of LASSO MLLR in Section 2. Tested on the 5k-WSJ0 task in an utterance-by-utterance unsupervised self-adaptation scenario, the proposed LASSO MLLR algorithms significantly outperform the standard MLLR alternatives. Moreover, LASSO MLLR also has very good interpretability on transformation matrix elements.

2. LASSO MLLR

In this section, we first give a brief review of maximum likelihood linear regression (MLLR). Then, LASSO MLLR is formulated as the standard MLLR with the L_1 constraint. Solution and implementation detail are given. Finally, we also propose LASSO-P MLLR from a Bayesian view.

2.1 Maximum Likelihood Linear Regression (MLLR)

In automatic speech recognition, acoustic model is used to characterize the likelihood of acoustic feature with respect to the underlying word sequence. The main stream of acoustic modeling is to use hidden Markov models (HMMs) which gracefully handle the dynamic time evolution of speech signals and characterize it as a parametric random process. Each state of HMMs usually has a continuous output density function modeled by a Gaussian mixture model (GMM):

$$p(x|\Lambda_i) = \sum_j c_j N(x; \mu_{ij}, \sigma_{ij}^2)$$

where x is the observation vector, $N(x; \mu_{ij}, \sigma_{ij}^2)$ is a Gaussian with mean vector μ_{ij} and variance vector σ_{ij}^2, and c_j is the weight for *j*-th Gaussian component in the GMM.

MLLR model adaptation is one way to address the mismatches between the training and testing environments. MLLR is performed with a linear transformation matrix W on the augmented *(p+1)*-dimension mean vector of each Gaussian in the ASR system as

$$\hat{\mu}_s = W\xi_s,$$

where *W* is a *p*(p+1)* matrix with $W = [b\ A]$, b is a bias vector, and *A* is a rotation matrix. $\hat{\mu}_s$ is the new mean of state *s* and ξ_s is the augmented vector of the mean vector μ_s.

$$\xi_s = [1, \mu_s']'.$$

This can be solved with the expectation maximization (EM) algorithm [8] by maximizing the auxiliary function:

$$Q_{MLLR} = -\frac{1}{2}\sum_i \sum_{t=1}^T \gamma_i(t)[plog2\pi + \log|\Sigma_i| + (x_t - W\xi_i)'\Sigma_i^{-1}(x_t - W\xi_i)]\ ,$$

or

$$Q_{MLLR} \sim -\frac{1}{2}\sum_i \sum_{t=1}^T \gamma_i(t)[(x_t - W\xi_i)'\Sigma_i^{-1}(x_t - W\xi_i)],$$

where x_t is the observation vector at time *t*, $\gamma_i(t)$ is the posterior probability of state i at time *t*, and Σ_i is the covariance matrix of state i.

2.2 LASSO MLLR and Its Solution

As stated in [1], linear regression can be viewed as a special case of MLLR, and it is straightforward to borrow the success of LASSO [7] to address the linear regression estimation problem in MLLR. In particular, we formulate LASSO MLLR as follows:

$$max -\frac{1}{2}\sum_i \sum_{t=1}^T \gamma_i(t)[(x_t - W\xi_i)'\Sigma_i^{-1}(x_t - W\xi_i)]$$

with a constraint that

$$\sum_{jk}|A_{jk}| \le c\ ,$$

where *c* is a positive constant, and A_{jk} is the element in the *j*-th row and *k*-th column of the rotation matrix *A*.

Re-formulating the problem in a Lagrange form, we have

$$\widehat{W}_{LASSO} = [\hat{b}\ \hat{A}\ ] = \text{arg}min\ \frac{1}{2}\sum_i \sum_{t=1}^T \gamma_i(t)[(x_t - W\xi_i)'\Sigma_i^{-1}(x_t - W\xi_i)] + \lambda \sum_{jk}|A_{jk}|^1 \quad (1)$$

where λ is the Lagrange parameter.

We note that there is no close form solution for Eq. (1) and we solve it in an iterative manner.

First we initialize $[b\ A\ ]$ from the standard MLLR solution, then update A by updating A_{jk} sequentially in each iteration. More specially, given the current estimates, A^* and b^*, let A_{-jk} be A^* with its (j,k) entry replaced by 0. Denote by

$$z_{tj} = x_{tj} - b_j - \left(A_{-jk}\right)_j \mu_i$$

and

$$w_{ijt} = \gamma_i(t)(\Sigma_i^{-1})_{jj}$$

Then we update A_{jk} by solving

$$\text{arg}min\ \frac{1}{2}\sum_i \sum_{t=1}^T w_{ijt}\left(z_{tj} - A_{jk}\mu_{ik}\right)^2 + \lambda \sum_{jk}|A_{jk}|$$

The solution can be computed explicitly from the first order condition:

$$0 = -\sum_i \sum_{t=1}^T w_{ijt}\mu_{ik}\left(z_{tj} - A_{jk}\mu_{ik}\right) + \lambda sign\left(A_{jk}\right) = -c + dA_{jk} + \lambda sign\left(A_{jk}\right)$$

where

$$c = \sum_i \sum_{t=1}^T w_{ijt}\mu_{ik}z_{tj} \quad (2)$$

$$d = \sum_i \sum_{t=1}^T w_{ijt}\mu_{ik}^2 \quad (3)$$

By Karush–Kuhn–Tucke theorems [9], we have

if $c \ge 0$,

$$A_{jk} = \frac{c}{d} - \frac{\lambda sign\left(A_{jk}\right)}{d} = \begin{cases} \frac{c}{d} - \frac{\lambda}{d} = \left(\frac{c}{d} - \frac{\lambda}{d}\right) sign\left(\frac{c}{d}\right) & if\ c \ge \lambda \\ 0 & otherwise \end{cases} \quad (4)$$

if $c < 0$,

$$A_{jk} = \frac{c}{d} - \frac{\lambda sign\left(A_{jk}\right)}{d} = \begin{cases} \frac{c}{d} + \frac{\lambda}{d} = \left(-\frac{c}{d} - \frac{\lambda}{d}\right) sign\left(\frac{c}{d}\right) & if\ -c \ge \lambda \\ 0 & otherwise \end{cases} \quad (5)$$

Combining Eqs. (4) and (5) together, we have the final solution:

$$A_{jk} = \left(\left|\frac{c}{d}\right| - \frac{\lambda}{d}\right)_+ sign\left(\frac{c}{d}\right) \quad (6)$$

After updating all entries of the matrix A, we then update b again by solving the first order condition:

$$b = (\sum_i \sum_t \gamma_i(t)\Sigma_i^{-1})^{-1} \sum_i \sum_t \gamma_i(t)\Sigma_i^{-1}(x_t - A\mu_i) \quad (7)$$

To summarize, the solution of LASSO MLLR can be computed as follows:

Step 1) Initialize $[b\ A]$ from last iteration. Use the standard MLLR solution in the first iteration.

Step 2) Update every element in matrix A with Eq. (6).

Step 3) Update the bias b with Eq. (7).

Step 4) Check whether the parameters are converged. If not, go to Step 1).

2.3 Implementation of LASSO MLLR

With simple reformulation for Eq. (2), we can have

$$c = \sum_i \sum_{t=1}^{T} w_{ijt}\mu_{ik}z_{tj} = c_1 - c_2 - c_3$$

with

$$c_1 = \sum_i \left(\sum_{t=1}^{T} \gamma_i(t)\, x_{tj} \right) (\Sigma_i^{-1})_{jj}\mu_{ik} \quad (8)$$

$$c_2 = \left\{ \sum_i \left(\sum_{t=1}^{T} \gamma_i(t) \right) (\Sigma_i^{-1})_{jj}\mu_{ik} \right\} b_j \quad (9)$$

$$c_3 = \sum_{l \neq k} \left\{ \sum_i \left(\sum_{t=1}^{T} \gamma_i(t) \right) (\Sigma_i^{-1})_{jj}\mu_{ik}\mu_{il} \right\} A_{jl} \quad (10)$$

Note that c_1 in Eq. (8) does not depend on the matrix, and can be computed in advance. The components inside the big brackets in Eqs. (9) and (10) are also invariant across iterations. In each iteration, they are multiplied by b_j and A_{jl}, respectively.

The d value in Eq. (3) also remains the same across iterations and can be computed in advance as

$$\begin{aligned} d &= \sum_i \sum_{t=1}^{T} \left(w_{ijt}\mu_{ik}^2 \right) \\ &= \sum_i \left(\sum_{t=1}^{T} \gamma_i(t) \right) (\Sigma_i^{-1})_{jj}\mu_{ik}^2 \end{aligned} \quad (11)$$

Therefore, all the accumulator statistics involved in Eqs. (2) and (3) can be pre-computed and do not change during the iterative update of A_{jk} . Only very limited multiplications are needed in Eqs. (9) and (10). This is also true for the update of bias b in Eq. (7). As a result, the computation cost is relatively low.

2.4 LASSO MLLR with Prior

We can also formulate LASSO MLLR with a maximum a *posterior* objective function as

$$Q_{MAP} = Q_{MLLR} + logP(A)$$

We assume every component of the rotation matrix A follows a Laplace distribution:

$$p(A) \propto \prod_{jk} \frac{1}{2V_{jk}} \exp\left[-\frac{|A_{jk} - M_{jk}|}{V_{jk}} \right]$$

where M_{jk} and V_{jk} are the hyper parameters for A_{jk}, and they can be estimated either from a training set or a development set. We can formulate the LASSO MLLR with priors (LASSO-P MLLR):

$$\begin{aligned} \widehat{W}_{LASSO-P} = [\hat{b}\ \hat{A}] = \\ \text{argmin}\ \tfrac{1}{2}\textstyle\sum_i \sum_{t=1}^{T} \gamma_i(t)[(x_t - \\ W\xi_i)'\Sigma_i^{-1}(x_t - W\xi_i)] + \sum_{jk} \frac{|A_{jk} - M_{jk}|}{V_{jk}} \end{aligned} \quad (12)$$

By comparing Eq. (12) with Eq. (1), we can see the LASSO-P MLLR shrinks component A_{jk} to its prior M_{jk} instead of zero value.

The solution of LASSO-P MLLR is very similar to Eq. (5):

$$A_{jk} = M_{jk} + \left(\left| \frac{c}{d} \right| - \frac{1}{dV_{jk}} \right)_+ sign\left(\frac{c}{d} \right) \quad (13)$$

with

$$c = \sum_i \sum_{t=1}^{T} w_{ijt}\mu_{ik}z_{tj} - dM_{jk}$$

and

$$d = \sum_i \sum_{t=1}^{T} w_{ijt}\mu_{ik}^2$$

3. Experiment

We used the 5k-WSJ0 task to evaluate the effectiveness of LASSO MLLR. Testing was conducted on the Nov92 evaluation set with 330 utterances. Baseline models used cross-word triphones obtained with maximum likelihood estimation. There were 2818 shared states resulted from a decision tree state clustering. Each state observation density is characterized by an 8-component Gaussian mixture model. The input 39-dimension features were 12 MFCCs + energy, and their first and second order time derivatives. A trigram language model was used for decoding. The baseline word error rate (WER) was 5.08%.

To evaluate the ASR performance with unsupervised self-adaptation, every test utterance was first decoded to get a hypothesized sequence of words. Then this decoded transcription was used to adapt models for this utterance. The adapted model is used to get the final decoding sentence.

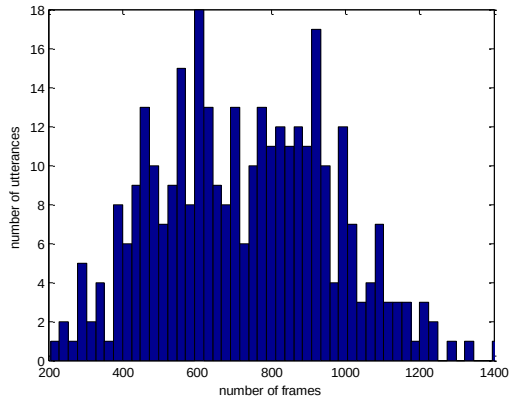

Figure 1. Histogram of the number of frames in the WSJ0 test utterances

As shown in Figure 1, the test set of WSJ0 is quite suitable to evaluate the proposed approach on the insufficient data problem. If we consider that roughly 39*39+39=1560 frames are need to reliably estimate the MLLR transform matrix, no test utterance can reach that criterion. Note that the unsupervised adaptation technology is a general technology in ASR, the discovery in this paper should be able to apply to even larger ASR tasks than 5k-WSJ0.

When estimating the LASSO MLLR matrix, we start from the estimate obtained with standard MLLR, and then iteratively update the transformation matrix by using the algorithm in Section 2.2.

Figure 2 shows the influence of λ for LASSO MLLR by plotting all the non-zero elements of the estimated transformation matrix A in a 39*39 image. If the resulting A_{jk} in LASSO MLLR is non-zero, there is a circle in the j-th row and k-th column. When λ increases from 20 to 60, more and more elements in matrix A become 0. Here are some observations.

- Most non-zero elements are in the upper triangle, which means the elements to the left of the j-th column ($A_{jk}, k \leq j$) contribute more to estimating the transformed GMM mean parameters of the j-th dimension.
- There is a clear diagonal line from the (0, 0) to (38, 38) coordinates, showing that A_{jj} is very important for the MLLR transformation. Moreover, the magnitude of A_{jj} is significantly larger than other values of A_{jk} $(k \neq j)$ which cannot be shown in the figures.
- There are three horizontal lines in the y-axis with values of 12, 25, and 38. They are corresponding to the feature dimensions of energy, delta energy, and delta delta energy, which are the most important feature dimensions.

Table 1 compares the WERs of the baseline, the standard MLLR, and the LASSO MLLR with different λ values. The standard MLLR gets a slightly better WER than the baseline. Within a range of λ (from 20 to 100), LASSO MLLR is much better than the standard MLLR. As expected if λ is too small (e.g., 1), LASSO MLLR behaves similarly as the standard MLLR, with few zero elements. In contrast, if λ is too large, the matrix A is very sparse and we begin to lose the power of prediction. The best LASSO MLLR achieved a 4.60% WER, corresponding to about 6.7% relative WER reduction from the 4.93% WER of the standard MLLR.

The WER of the LASSO-P MLLR is also given in Table 1. The hyper parameters, M_{jk} and V_{jk}, are estimated from the MLLR transformation matrices for every speaker in the training set.

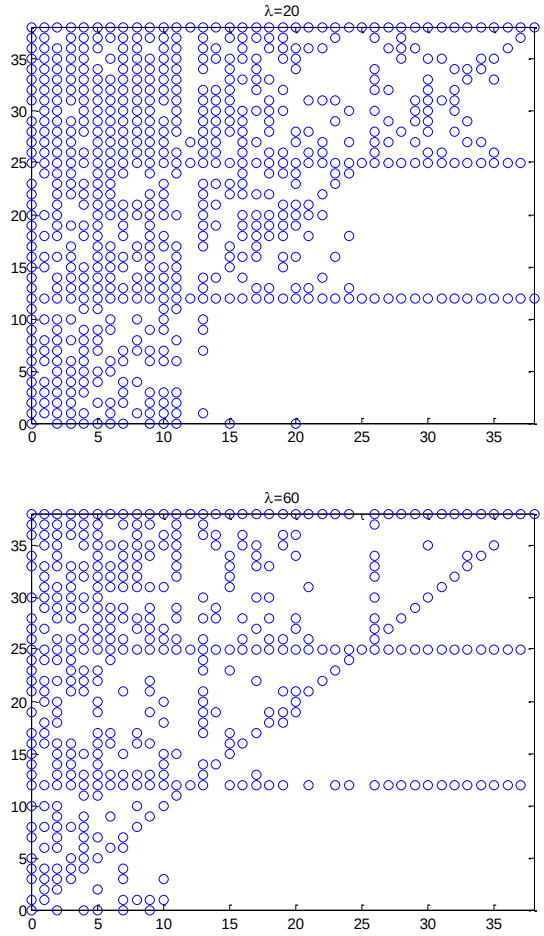

Figure 2. Non-zero elements in LASSO MLLR matrix A. If A_{jk} is non-zero, there is a circle in the j-th row and k-th column.

As shown in Figure 2, by adjusting the value of λ, we can easily control the sparsity of the transformation matrix. This is a natural way to address the sparse data problem. LASSO MLLR also offers a good interpretation of the

transformation coefficients as we have demonstrated before. By examining plots, such as those in Figure 2, we can have a good sense of what reasonable λ values to be set in Eq. (1) if we desire a specific set of constraints for the MLLR matrix.

Table 1. Detailed WERs of baseline, standard MLLR, LASSO MLLR with different λ values, and LASSO-P MLLR

system	WER
Baseline	5.08
MLLR	4.93
LASSO MLLR ($\lambda = 1$)	4.84
LASSO MLLR ($\lambda = 20$)	4.71
LASSO MLLR ($\lambda = 60$)	4.60
LASSO MLLR ($\lambda = 100$)	4.69
LASSO-P MLLR	4.67

4. Conclusion and future work

In this study we have proposed LASSO MLLR model adaptation to address the data insufficiency problem. Because the L_1 constraint can make some transformation elements be exactly zero, LASSO MLLR is a natural solution to the adaptation data insufficiency problem. We used the 5k-WSJ0 task for the unsupervised self-adaptation test. LASSO MLLR achieved about 6.7% relative WER reductions from standard MLLR. Although a better WER than ridge MLLR was not clearly observed, the proposed LASSO MLLR has better parameter interpretability, and provides a natural way to address the data sparseness issue.

This paper only presents our initial study. We are now working on some potential improvements of LASSO MLLR. For example, in the current set of experiments, we fix the λ value for all testing utterances. It is desirable to automatically set the λ value according to the amount of available adaptation data. In some application scenarios, adapting model for every utterance is not affordable. We need to extend the LASSO adaptation to constrained MLLR [9], which can apply the transformation to the feature space.

References

[1] Leggetter, C. J. and Woodland, P. C. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, vol. 9, no. 2, pp. 171-185, 1995.

[2] Gauvain, J. -L. and Lee, C. -H. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. on Speech and Audio Proc.*, vol. 2, no. 2, pp. 291-298, 1994.

[3] Wang, Y. Y., Yu, D., Ju, Y. -C., and Acero, A. An introduction to voice search. in *IEEE Signal Processing Magazine (Special Issue on Spoken Language Technology)*, pp. 29-38, May 2008.

[4] Chesta, C., Siohan, O., and Lee, C. -H. Maximum a posteriori linear regression for hidden Markov model adaptation. in *Proc. Eurospeech*, pp. 211-214, 1999.

[5] Kuhn, R., Junqua, J. -C., Nguyen, P., and Niedzielski, N. Rapid speaker adaptation in eigenvoice space. *IEEE Trans. on Speech and Audio Proc.*, Vol. 8, No. 6, pp. 695-707, 2000.

[6] Chen, K. -T., Liau, W. -W., Wang, H. -M., and Lee, L. -S. Fast speaker adaptation using eigenspace-based maximum likelihood linear regression. in *Proc. ICSLP*, vol. 3, pp. 742-745, 2000.

[7] Tibshirani, R. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society*, Series B, vol. 58, pp.267–288, 1996.

[8] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 39(1):1–38, 1977.

[9] Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, 2000.

[10] Gales, M. J. F. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, vol. 12, pp. 75-98, 1998.