Automatic Estimation of Decoding Parameters Using Large-Margin Iterative Linear Programming

Brian Mak and Tom Ko

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
{mak,tomko}@cse.ust.hk

Abstract

The decoding parameters in automatic speech recognition — grammar factor and word insertion penalty — are usually determined by performing a grid search on a development set. Recently, we cast their estimation as a convex optimization problem, and proposed a solution using an iterative linear programming algorithm. However, the solution depends on how well the development data set matches with the test set. In this paper, we further investigates an improvement on the generalization property of the solution by using large margin training within the iterative linear programming framework. Empirical evaluation on the WSJ0 5K speech recognition tasks shows that the recognition performance of the decoding parameters found by the improved algorithm using only a subset of the acoustic model training data is even better than that of the decoding parameters found by grid search on the development data, and is close to the performance of those found by grid search on the test set.

Index Terms: discriminative training, convex optimization, large margin training, iterative linear programming, ranking support vector machine.

1. Introduction

In the statistical pattern classification framework, the *maximum a posteriori* (MAP) decision rule is used in automatic speech recognition (ASR) to determine the recognized output. That is, given a sequence of T acoustic observations, $\mathbf{x}_1^T = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\}$, the objective of ASR is to find an N-word sequence, $\hat{\mathbf{w}}_1^N = \{\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \cdots, \hat{\mathbf{w}}_N\}$, such that

$$\hat{\mathbf{w}}_1^N = \underset{\mathbf{w}_1^N, N}{\operatorname{argmax}} \; p(\mathbf{w}_1^N | \mathbf{x}_1^T) \tag{1}$$

Applying the Bayes Theorem and expressing quantities in the log domain, Eqn.(1) may be rewritten as

$$\hat{\mathbf{w}}_1^N = \underset{\mathbf{w}_1^N, N}{\operatorname{argmax}} \; \underbrace{\ln p(\mathbf{x}_1^T | \mathbf{w}_1^N)}_{\text{acoustic score}} + \underbrace{\ln p(\mathbf{w}_1^N)}_{\text{language score}} \; . \tag{2}$$

The first term is called the acoustic score that is computed from acoustic models, and the second term is called the language score which is commonly computed from n-gram language models.

This research is supported by the Research Grants Council of the Hong Kong SAR under the grant numbers 617406 and 617008.

Despite the strong theoretical foundation of the MAP decision rule, a direct application of Eqn.(2) to ASR gives poor results in practice. There can be two reasons: (1) the particular mathematical models used in acoustic modeling and language modeling may not be correct, and (2) the dynamic ranges of the acoustic scores and language scores are very different. It is found that ASR performance can be greatly improved if the two scores are properly balanced with decoding parameters before making the MAP decision [1]. In general, the ASR objective function is modified as

$$\hat{\mathbf{w}}_1^N = \underset{\mathbf{w}_1^N, N}{\operatorname{argmax}} \Big\{ \ln p(\mathbf{x}_1^T | \mathbf{w}_1^N) + K_{gf} \ln p(\mathbf{w}_1^N) + K_{wip} N \Big\} \; , \tag{3}$$

where K_{gf} and K_{wip} are the decoding parameters, and they are called grammar factor and word insertion penalty respectively. The decoding parameters are commonly hand-tuned by performing a grid search using utterances from a development set (that is separate from the training set of the acoustic models). They may also be estimated automatically by discriminative model combination [2], MAP training [3], or using some heuristics [4].

Table 1: WSJ0 recognition performance on the standard test set using decoding parameters estimated from various data sets and by different approaches.

Training Set (#Utt.)	Method	Word (Utt.) Acc. (%)
test set (330)	grid search	93.16 (44.55)
dev subset (442)	grid search	92.92 (44.55)
test set (164, 166)	ILP	92.88 (43.64)
dev set (442)	ILP	92.53 (42.42)
train subset (1175)	ILP	91.72 (37.58)

Recently we proposed an iterative linear programming (ILP) algorithm for finding the optimal values of the parameters in a linear function [5]. It is shown also effective for the automatic estimation of the decoding parameters [6] in Eqn. (3). Nevertheless, there is still a performance gap between the decoding parameters found by our ILP algorithm and those found by a brute-force grid search. For example, in the Wall Street Journal WSJ0 5K task, we compare the performance between the decoding parameters found by grid search and those found by our ILP using the test data[1] or development data; the results

[1]This is a cheating experiment in which the WSJ0 test data were

are shown in Table 1. The results seems to agree with the general belief that it is better to train the decoding parameters with data that are as similar to the test set as possible. In any case, there is still a performance gap between the solutions of ILP and grid search: about 0.6% (absolute) if the development set is used, and about 2% if one only has the acoustic model training data.

In this paper, we investigate to improve the generalization property of ILP by large margin training [7]: it is not enough that the correct word sequence beats all other hypotheses; it also has to do it by a margin as large as possible. We will show that ILP with large margin training together may close the performance gap if the margin is properly chosen.

This paper is organized as follows. In Section 2, we will review our iterative LP algorithm for the optimization of the decoding parameters, and describe the modification for large margin training. That is followed by experimental evaluation in Section 3 and conclusions in Section 4.

2. Large-Margin Iterative Linear Programming (LMILP)

We will first review our previous iterative linear programming (ILP) method for the estimation of decoding parameters using the following notations:

L : no. of training utterances
J : no. of competing hypotheses for each utterance
$\mathbf{x}_i$: the ith training utterance
$\hat{\mathbf{w}}_i$: correct transcription of $\mathbf{x}_i$ with N_i words
$\mathbf{w}_{ij}$: jth competing hypothesis of $\mathbf{x}_i$ with N_{ij} words

Notice that we have dropped the duration and length specification from the acoustic and word sequences for simplicity.

2.1. Review of Iterative Linear Programming

For each of the L training utterance $\mathbf{x}_i$, we define a discriminant d_{ij} with respect to each of its J competing hypotheses. d_{ij} is the difference between the recognition score of its correct word sequence $\hat{\mathbf{w}}_i$ and its jth competing word sequences $\mathbf{w}_{ij}$, and is defined as follows:

$$\begin{aligned}\forall i, \forall j, \quad & d_{ij} \\ = \; & (\ln p(\mathbf{x}_i|\hat{\mathbf{w}}_i) + K_{gf} \ln p(\hat{\mathbf{w}}_i) + K_{wip} N_i) \\ & - (\ln p(\mathbf{x}_i|\mathbf{w}_{ij}) + K_{gf} \ln p(\mathbf{w}_{ij}) + K_{wip} N_{ij}) \\ = \; & u_{ij} + K_{gf} v_{ij} + K_{wip} z_{ij} \; . \end{aligned} \tag{4}$$

where

$$\begin{aligned} u_{ij} &= \ln p(\mathbf{x}_i|\hat{\mathbf{w}}_i) - \ln p(\mathbf{x}_i|\mathbf{w}_{ij}) \\ v_{ij} &= \ln p(\hat{\mathbf{w}}_i) - \ln p(\mathbf{w}_{ij}) \\ z_{ij} &= N_i - N_{ij} \; . \end{aligned}$$

For LP optimization of the decoding parameters, we would like these discriminants to be positive so that the true word sequences always prevail during recognition.

$$\forall i, \forall j, \quad u_{ij} + K_{gf} v_{ij} + K_{wip} z_{ij} \geq 0 \; . \tag{5}$$

divided to two equal halves, and then one subset was used for finding the optimal decoding parameters by ILP and its solution was used to decode the other subset. The experiment was repeated by swapping the role of the two subsets, and their recognition results were combined for reporting.

However, in general, not all of the $L \times J$ constraints can be satisfied. We may relax the requirements by introducing *slack variables* $\xi_{ij} \geq 0$ into the constraints, and require

$$\forall i, \forall j, \quad u_{ij} + K_{gf} v_{ij} + K_{wip} z_{ij} + \xi_{ij} \geq 0 \; . \tag{6}$$

2.1.1. LP Formulation

The slack variables implements the hinge loss function so that their values for correctly recognized utterances are zero, and their values for incorrectly recognized utterances are positive. One may interpret the slack variables in Eqn. (6) as an approximate measure of the string-level utterance recognition errors, and tries to minimize the sum of these slack variables over all training utterances and their competitors. Thus, we formulate the estimation of the decoding parameters as a standard LP problem that minimizes the approximate utterance errors as follows:

$$\min_{K_{gf}, K_{wip}} \sum_i \sum_j \xi_{ij} \tag{7}$$

subject to the following constraints

$$\forall i, \; \forall j, u_{ij} + K_{gf} v_{ij} + K_{wip} z_{ij} + \xi_{ij} \geq 0 \; , \tag{8}$$

$$\forall i, \; \forall j, \xi_{ij} \geq 0 \; , \tag{9}$$

$$K_{gf} \geq 0 \; . \tag{10}$$

2.1.2. Iterative LP

In theory, LP is a convex optimization problem and the solution is globally optimal (with respect to the feasible region). However, in our problem, the feasible region is, in any practical sense, infinite because there are practically infinite possible competing word sequences with infinite possible alignments! As we may only have a finite set of competing hypotheses (for example, N-best hypotheses), we would not have a complete knowledge of the feasible region. As a consequence, the globally optimal solution found in the incomplete feasible region is unlikely the solution of our original intended problem (which assumes complete knowledge of the feasible region), and we do not want to move directly to that solution.

In [6], we propose two modifications to the standard LP:

- impose additional constraints on the decoding parameters so that they are not allowed to change from their current values too much as follows:

$$|K_{gf}(n+1) - K_{gf}(n)| \leq \Delta K_{gf_{max}} \tag{11}$$

$$|K_{wip}(n+1) - K_{wip}(n)| \leq \Delta K_{wip_{max}} \tag{12}$$

  where n is the iteration count.

- repeat the LP procedure several times. At the end of an iteration, the LP solution is used to re-estimate the N-best hypotheses, and the LP estimation of the decoding parameters is repeated.

The iterative LP algorithm stops when a pre-specified maximum number of iterations n_{max} is reached, or when the relative change of the decoding parameters $\sqrt{K_{gf}(n)^2 + K_{wip}(n)^2}$ is smaller than a convergence threshold θ.

Furthermore, we tie the slack variables ξ_{ij} of all competing hypotheses for a training utterance, say, $\mathbf{x}_i$, together to a single slack variable ξ_i. The tying in effect implements a min-max cost function for the LP problem.

2.2. Large Margin Training

One way to improve the generalization of a classifier is to require the classifier to have a large margin [7] despite of an increase in training errors. It is a regularization method to avoid overfitting of the training data. In this paper, we introduce large margin training to our iterative linear programming algorithm by requiring the recognition score of the correct word sequence $\hat{\mathbf{w}}_i$ of an utterance $\mathbf{x}_i$ to be greater than that of any of its competing word sequences $\mathbf{w}_{ij}$ by a positive margin $M \geq 0$. That is, we modify Eqn. (6) as follows:

$$\forall i, \forall j, \quad u_{ij} + K_{gf} v_{ij} + K_{wip} z_{ij} + \xi_{ij} \geq M \ . \qquad (13)$$

As we will see in Section 3, although the change seems small, it is very effective.

3. Experimental Evaluation on WSJ0

The proposed large-margin iterative linear programming algorithm (LMILP) was evaluated on the Wall Street Journal WSJ0 5K tasks.

3.1. WSJ0 Corpus and Acoustic Modeling

The standard SI-84 training set was used for training the speaker-independent (SI) model. It consists of 83 speakers (41 male speakers and 42 female speakers) and 7,138 utterances for a total of about 14 hours of training speech. The standard Nov'92 5K non-verbalized test set was used for evaluation using the standard 5K-vocabulary bigram which has a perplexity of 111. It consists of 8 speakers (5 male and 3 female speakers), each with about 40 utterances.

The SI model consists of 15,449 cross-word triphones based on 39 base phonemes. Each triphone model is a strictly left-to-right 3-state continuous-density hidden Markov model (CDHMM), with a Gaussian mixture density of 16 components per state, and there are totally 3,132 tied states. In addition, there are a 1-state short pause model and a 3-state silence model. The traditional 39-dimensional MFCC vectors were extracted at every 10ms over a window of 25ms.

"Optimal" decoding parameters were found by an extensive grid search using the test set and a subset of the si_dt_05 development set. The baseline recognition performances with these optimal decoding parameters are given in Table 1.

3.2. Experimental Setup for LMILP

Two data sets were used in the estimation of the decoding parameters by LMILP:

- a subset of the standard WSJ0 training set, consisting of $L = 1175$ utterances and 83 speakers.
- a subset of the standard WSJ0 development set si_dt_05, consisting of $L = 442$ utterances and 10 speakers.

The reason for subsetting the two data sets is that not all words in the remaining utterances are covered by the WSJ0 bigram language model. The development subset was also used in the grid search result in Table 1. Unless otherwise stated, all reported results used the WSJ0 training subset.

Various parameters used in the LMILP were set as follows.

- the competing hypotheses were found by N-best decoding with $N = 20$.
- the LP problems were solved by the Mosek optimization software[2].
- the maximum number of iterations $n_{max} = 10$.
- the convergence threshold $\theta = 10^{-4}$.
- 4 arbitrarily chosen starting points (K_{gf}, K_{wip}) were tried: (0, 0), (20, -20), (0, -20), and (20, 20).
- For the starting points (0, 0) and (20, -20), $\Delta K_{gf_{max}} = 7$ and $\Delta K_{wip_{max}} = 10$; for the starting points (0, -20) and (20, 20), $\Delta K_{gf_{max}} = 15$ and $\Delta K_{wip_{max}} = 30$.

[2]http://www.mosek.com

Table 2: Effect of the margin value on LMILP. (The word or utterance recognition accuracies with $M \geq 70$ are statistically significantly better than those with $M = 0$ at the 95% confidence level.)

M	Parameters		Word (Utt.) Acc.(%)		
	K_{gf}	K_{wip}	Train Subset	Dev. Subset	Test Set
0	9.79	-7.23	97.00 (71.83)	90.88 (39.82)	91.72 (37.58)
1	9.81	-7.34	97.00 (71.83)	90.88 (39.82)	91.72 (37.58)
10	9.90	-8.60	96.99 (71.74)	91.03 (40.50)	91.74 (37.58)
50	12.32	-11.93	96.98 (71.40)	91.86 (43.21)	92.55 (42.12)
70	13.70	-12.95	96.98 (71.32)	91.97 (44.12)	92.85 (44.55)
80	14.32	-13.95	96.92 (71.40)	91.92 (44.34)	92.94 (44.85)
100	15.20	-15.57	96.87 (71.06)	91.66 (42.76)	93.05 (44.85)
500	16.43	-17.85	96.76 (70.04)	91.64 (42.31)	93.03 (45.15)
1000	16.43	-17.85	96.76 (70.04)	91.64 (42.31)	93.03 (45.15)
∞	16.43	-17.85	96.76 (70.04)	91.64 (42.31)	93.03 (45.15)

3.3. Effect of Margin

Using 1175 utterances of the WSJ0 training subset, the decoding parameters were estimated using the proposed LMILP algorithm with increasing margin M, starting at (K_{gf}, K_{wip}) = (0, 0), and the solutions were used to decode the utterances in the various WSJ0 data sets. The detailed results are shown in Table 2 and the utterance accuracies are plotted in Fig. 1. From Fig. 1, it is clear that when the margin increases, the utterance accuracy of the training subset decreases (or the training error increases); there is a tradeoff between generalization (with a greater margin) and the training error. The benefit of generalization can be seen from the recognition performance on the development subset and the test set which are unseen during LMILP optimization: the performance generally improves with a larger margin. Although there is a small peak on the development subset, it is fair to say that both data sets prefer the margin as large as possible.

We analyze the difference in the recognition score between the correct word sequence and the recognized sequence for each of the 1175 utterances in the training subset before and after the introduction of large-margin training. Fig. 2 shows the distribution of the recognition score differences when the decoding parameters were estimated by our previous ILP (i.e., with no margin, $M = 0$) and our new LMILP with a large margin of $M = 1000$. It is observed that the winners win by a larger margin, and so do the losers.

3.4. Convergence with Different Starting Points

We also checked the convergence of the LMILP algorithm when it started with 4 arbitrary initial points with the margin $M = 80$. The result is shown in Fig. 3 and Fig. 4. In all four cases, the algorithm converged in about 5 iterations to give the same optimal decoding parameters $(K_{gf}, K_{wip}) = (14.32, -13.95)$.

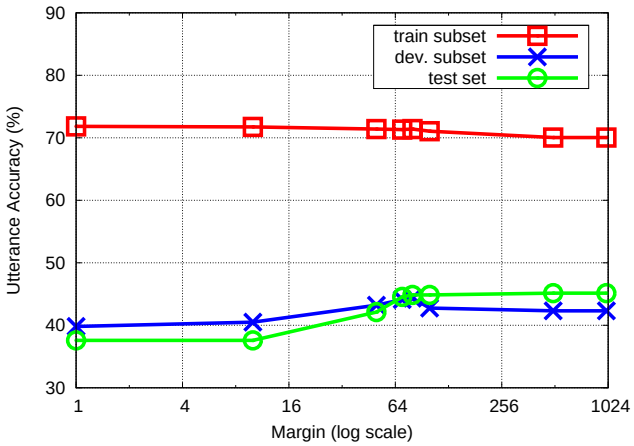

Figure 1: Effect of margin on LMILP.

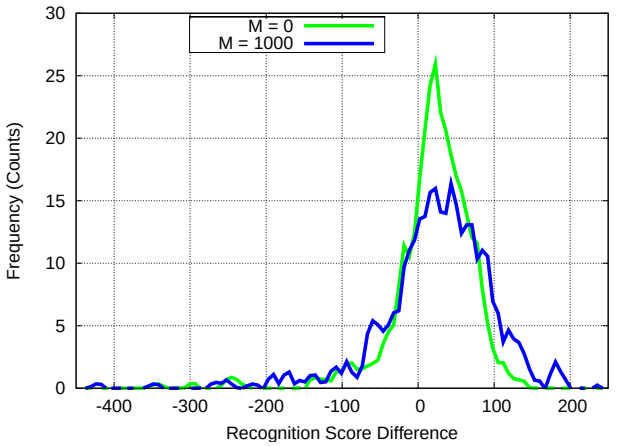

Figure 2: Distribution of the recognition score differences between the correct transcription and the top-best hypothesis before and after large-margin training.

4. Conclusions

This paper investigates the effect of large margin training on the estimation of decoding parameters using iterative linear programming (ILP). Large margin training lives up to its promise of improving the generalization of the solution given by ILP. From Table 1 and Table 2, we can see that for this problem, it is better to use a margin as large as possible and it is achieved using only a subset of the acoustic model training data; no additional data are needed. The decoding parameters thus obtained give a word accuracy of 93.03% on the WSJ0 test set, which is even better than the accuracy (92.92%) produced by the decoding parameters found by grid search using the development data, and is only 0.13% worse than the accuracy (93.16%) produced by the decoding parameters found by grid search using the test data.

5. References

[1] L. R. Bahl, R. Bakis, F. Jelinek, et al., "Language-model/acoustic channel balance mechanism," *IBM Technical Disclosure Bulletin*, vol. 23, no. 7B, pp. 3464–3465, Dec. 1980.

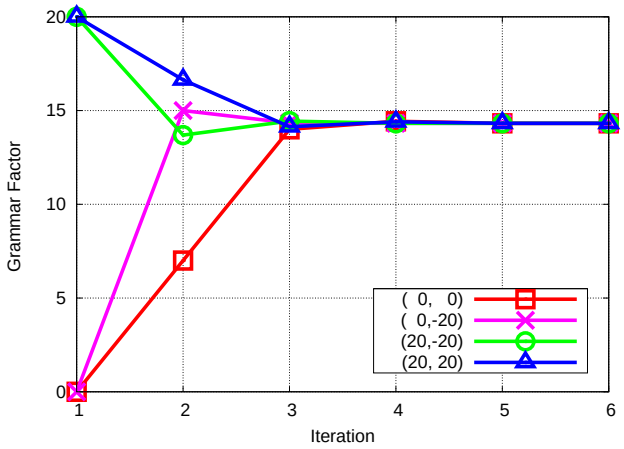

Figure 3: Large-margin iterative LP optimization of the grammar factor on WSJ0 using various initial (K_{gf}, K_{wip}) values.

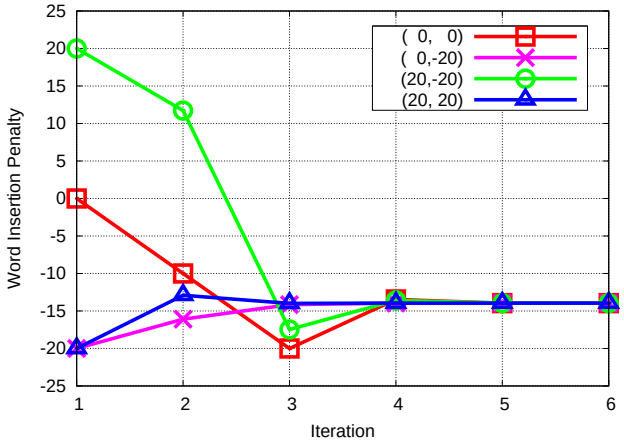

Figure 4: Large-margin iterative LP optimization of the word insertion penalty on WSJ0 using various initial (K_{gf}, K_{wip}) values.

[2] P. Beyerlein, "Discriminative model combination," in *Proc. of ICASSP*, 1998, pp. 481–484.

[3] Tadashi Emori, Yoshifumi Onishi, and Koichi Shinoda, "Automatic estimation of scaling factors among probabilistic models in speech recognition," in *Proc. of Interspeech*, 2007, pp. 1453–1456.

[4] T. Colthurst, T. Arvizo, C.-L. Kao, O. Kimball, S. Lowe, D. Miller, and J. V. Sciver, "Parameter tuning for fast speech recognition," in *Proc. of Interspeech*, 2007, pp. 1453–1456.

[5] Brian Mak and Benny Ng, "Discriminative training by iterative linear programming optimization," in *Proc. of ICASSP*, 2008, pp. 4061–4064.

[6] Brian Mak and Tom Ko, "Min-max discriminative training of decoding parameters using iterative linear programming," in *Proc. of Interspeech*, 2008, pp. 915–918.

[7] A.J. Smola, P.J. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., *Advances in Large Margin Classifiers*, MIT Press, Cambridge, 1998.