interspeech 2008 incorporating SST 08

Min-max Discriminative Training of Decoding Parameters Using 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

In automatic speech recognition, the decoding parameters — grammar factor and word insertion penalty — are usually hand-tuned to give the best recognition performance. This paper investigates an automatic procedure to determine their values using an iterative linear programming (LP) algorithm. LP naturally implements discriminative training by mapping linear discriminants into LP constraints. A min-max cost function is also defined to get more stable and robust result. Empirical evaluations on the RM1 and WSJ0 speech recognition tasks show that decoding parameters found by the proposed algorithm are as good as those found by a brute-force grid search; their optimal values also seem to be independent of the initial values set to start the iterative LP algorithm.

Index Terms: iterative linear programming, discriminative training, decoding parameters, min-max optimization.

1. Introduction

Current state-of-the-art automatic speech recognition (ASR) systems employ statistical pattern recognition to decode an utterance into a sequence of words. Specifically, the *maximum a posteriori* approach is used: given a sequence of T acoustic observations, $\mathbf{x}_1^T = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\}$, we would like to find the corresponding N-word sequence, $\hat{\mathbf{w}}_1^N = \{\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \cdots, \hat{\mathbf{w}}_N\}$, such that

$$\begin{aligned}\hat{\mathbf{w}}_1^N &= \underset{\mathbf{w}_1^N, N}{\operatorname{argmax}}\ p(\mathbf{w}_1^N|\mathbf{x}_1^T) \\ &= \underset{\mathbf{w}_1^N, N}{\operatorname{argmax}}\ p(\mathbf{x}_1^T|\mathbf{w}_1^N)p(\mathbf{w}_1^N) \\ &= \underset{\mathbf{w}_1^N, N}{\operatorname{argmax}}\ \underbrace{\ln p(\mathbf{x}_1^T|\mathbf{w}_1^N)}_{\text{acoustic score}} + \underbrace{\ln p(\mathbf{w}_1^N)}_{\text{language score}} \quad . \quad (1)\end{aligned}$$

The acoustic score is computed from acoustic models which are usually continuous density hidden Markov models (CDHMM), whereas the language score is computed from language models which are usually n-grams. Although the Bayesian decision rule of Eqn.(1) is correct in theory, in practice, there are two problems:

- the particular generative mathematical models used by an ASR system to capture the statistics of acoustics and linguistics may not be correct, and
- the dynamic ranges of the acoustic score and language score are usually very different. This is particularly important when CDHMMs are used since evaluation of state probability density functions in CDHMMs does not result in true probability quantities, while the evaluation of language model usually does.

A common heuristic used by the ASR community is to balance the two kinds of scores linearly with the introduction of two decoding parameters [1]: a grammar factor K_{gf}, and a word insertion penalty K_{wip}, and Eqn.(1) is re-written as

$$\hat{\mathbf{w}}_1^N = \underset{\mathbf{w}_1^N, N}{\operatorname{argmax}} \left\{ \ln p(\mathbf{x}_1^T|\mathbf{w}_1^N) + K_{gf} \ln p(\mathbf{w}_1^N) + K_{wip} N \right\}. \quad (2)$$

In practice, the two decoding parameters are usually hand-tuned using utterances from a development set. Nevertheless, there were a few attempts to automate their estimation. The most notable work perhaps is discriminative model combination (DMC) [2] proposed by Beyerlein. DMC is a variant of minimum-classification-error (MCE) discriminative training that minimizes the word error rate (WER) when several models (including acoustic and language models) are combined. Emori *et al.* treated Eqn.(2) as a log-linear model consisting of the decoding parameters, and estimated them directly using a gradient-ascent method [3]. On the other hand, Colthurst *et al.* devised a heuristic algorithm [4] to tune all system parameters (including grammar factor and word insertion penalty) in an ASR system with a cost function that trades off between WER and decoding time. Notice that both Emori's and Colthurst's methods are not discriminative.

Recently we proposed an iterative linear programming (LP) algorithm for discriminative training. The new algorithm was shown effective in the estimation of state-dependent stream weights for a multi-stream HMM system [5]. It turns out that the new algorithm can be used to determine the optimal values of the parameters in any linear function. In this paper, we will show that it is also effective in the estimation of the decoding parameters of Eqn. (2). There are several advantages of using our iterative LP algorithm:

- It is discriminative in nature.
- Unlike some other discriminative training methods, our cost function is linear.
- LP optimization is well studied with established solutions, and efficient LP solvers are freely available.
- Few parameters to tune.

This research is supported by the Research Grants Council of the Hong Kong SAR under the grant numbers DAG05/06.EG43, HKUST617406, HKUST617507, and HKUST617008.

- Scientists may concentrate on how to map a discriminative training problem into the new iterative LP algorithm, and do not have to worry about how to solve it.

2. Formulation of the Estimation of Decoding Parameters as an LP Problem

Suppose there are M training utterances $\mathbf{x}_i, i = 1, \ldots, M$, and their corresponding word transcriptions of length N_i are $\hat{\mathbf{w}}_i, i = 1, \ldots, M$. Further assume that we can identify J competing hypotheses for each training utterance $\mathbf{x}_i$, and its jth competitor of length N_{ij} is denoted as $\mathbf{w}_{ij}, j = 1, \ldots, J$. Notice that we have dropped the duration and length specification from the acoustic and word sequences for simplicity.

2.1. Linear Discriminants as LP Constraints

For each training utterance $\mathbf{x}_i$, we would like to have the recognition score of its correct word sequence $\hat{\mathbf{w}}_i$ greater than that of any of its competing word sequences $\mathbf{w}_{ij}$. Thus, we may have the following discriminants:

$$\begin{aligned}\forall i, \forall j, \quad & d_{ij} \\ = \; & (\ln p(\mathbf{x}_i|\hat{\mathbf{w}}_i) + K_{gf} \ln p(\hat{\mathbf{w}}_i) + K_{wip} N_i) \\ & - (\ln p(\mathbf{x}_i|\mathbf{w}_{ij}) + K_{gf} \ln p(\mathbf{w}_{ij}) + K_{wip} N_{ij}) \\ = \; & (\ln p(\mathbf{x}_i|\hat{\mathbf{w}}_i) - \ln p(\mathbf{x}_i|\mathbf{w}_{ij})) + K_{gf} \, (\ln p(\hat{\mathbf{w}}_i) \\ & - \ln p(\mathbf{w}_{ij})) + K_{wip} \, (N_i - N_{ij}) \; . \end{aligned} \tag{3}$$

Applying the following variable substitutions:

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

Eqn. (3) may be simplified as

$$\forall i, \forall j, \quad d_{ij} = u_{ij} + K_{gf} v_{ij} + K_{wip} z_{ij} \; . \tag{4}$$

Now, the discriminative training of the decoding parameters may be turned into a linear programming (LP) optimization problem by mapping each linear discriminant to an LP constraint as follows:

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

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

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

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.

2.2. LP Formulation

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. Using the constraints in Eqn. (6), we may, thus, formulate the estimation of the decoding parameters as a standard LP problem as follows:

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

subject to the following constraints

$$\begin{aligned} \forall i, \; \forall j, u_{ij} + K_{gf} v_{ij} + K_{wip} z_{ij} + \xi_{ij} &\geq 0 \; , & (8) \\ \forall i, \; \forall j, \xi_{ij} &\geq 0 \; , & (9) \\ K_{gf} &\geq 0 \; . & (10) \end{aligned}$$

Algorithm 1: An iterative linear programming algorithm for estimating the decoding parameters.

Step 0. Set the iteration index $n = 0$, and determine

- the initial values of the grammar factor $K_{gf}(0)$ and word insertion penalty $K_{wip}(0)$.
- the maximum change allowed in the two decoding parameters: $\Delta K_{gf_{max}}$ and $\Delta K_{wip_{max}}$.
- the maximum number of iterations n_{max}.
- convergence measure θ.

Step 1. Perform N-best decoding for each training utterance using the current decoding parameters, $K_{gf}(n)$ and $K_{wip}(n)$, and register the acoustic score difference u_{ij}, language score difference v_{ij}, and the difference in the number of words z_{ij} for each of the J hypotheses.

Step 2. Construct the linear programming problem of Eqns. (7–10) with the following additional constraints:

$$\begin{aligned} |K_{gf}(n+1) - K_{gf}(n)| &\leq \Delta K_{gf_{max}} & (11) \\ |K_{wip}(n+1) - K_{wip}(n)| &\leq \Delta K_{wip_{max}} & (12) \end{aligned}$$

Step 3. Solve the linear programming problem of Step 2.

Step 4. If the relative change of $\sqrt{K_{gf}(n)^2 + K_{wip}(n)^2}$ is less than the threshold θ, or n_{max} is reached, stop.

Step 5. Set $n = n + 1$, and go to Step 1.

2.3. Min-max Iterative Training Approach

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! The reason is that for any utterance, there are infinite number of word sequences of various lengths and with various amount of leading/trailing/embedding silences/pauses that can be considered as a competing hypothesis of the correct transcription. Consequently, in practice, one may only settle with using a subset of all possible competing hypotheses to approximate the feasible region. In our case, the competing hypotheses are generated by N-best decoding using the current values of the decoding parameters. Furthermore, since the current values of the decoding parameters may not be optimal (otherwise, we already have solved the problem), the competing hypotheses found by N-best decoding based on them, in general, will not give the same feasible region of our original intended LP problem.

We modify the iterative linear programming algorithm we previously proposed for stream weights estimation [5] for the current task as shown in Algorithm 1. The basic idea is that for an LP problem with an incomplete and approximate feasible region at hand, the globally optimal solution is unlikely the solution of our original intended problem (which assumes complete

knowledge of the feasible region), and we should not let the estimating parameters to move directly to that solution. Instead, additional constraints are imposed on the grammar factor and word insertion penalty so that they are not allowed to change from their current values more than $\Delta K_{gf_{max}}$ and $\Delta K_{wip_{max}}$ respectively in each iteration. Thus, an original one-step LP problem is turned into a sequence of LP problems with approximate feasible regions. By carefully controlling $\Delta K_{gf_{max}}$ and $\Delta K_{wip_{max}}$ in each iteration, it is hoped that the decoding parameters will converge gradually to their locally (if not globally) optimal values.

We further 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. Besides reducing the number of variables in the LP problem (which can result in substantial computational savings for large LP problems), the tying also indirectly implements a min-max cost function for the LP problem. That is, it tries to minimize the distance of the correct transcription from its strongest competitor for each training utterance. In our preliminary experiments, tying the slack variables in this way gives more stable performance with faster convergence.

Table 1: Recognition performance using the decoding parameters found by grid search on the test data.

Task	Word (Utt.) Accuracy	K_{gf}	K_{wip}
RM1	93.16% (69.00%)	5	0
WSJ0	93.16% (44.55%)	15	-30

3. Experimental Evaluation

The proposed iterative linear programming algorithm was evaluated on the Resource Management RM1 and Wall Street Journal WSJ0 5K tasks. The algorithm was run with the following 4 different initial values: { (K_{gf}, K_{wip}) } = { (0, 0), (0, 20), (20, 0), (20, 20) } to investigate its dependency on the initial condition.

Extensive grid search was also performed on the test data of each task to find the best decoding parameters. The results are shown in Table 1. The results give an "approximate" upper bound for our proposed estimation method.

Here are some common operations and experimental settings for both tasks:

(a) Feature extraction: the traditional 39-dimensional MFCC vectors were extracted at every 10ms over a window of 25ms.

(b) Acoustic modeling: each phonetic model was a strictly left-to-right 3-state continuous-density hidden Markov model (CDHMM). In addition, there were a 1-state short pause model and a 3-state silence model.

(c) System settings:

- the competing hypotheses were found by N-best decoding with $N = 20$.
- the LP problems were solved by the Mosek optimization software[1].
- $\Delta K_{gf_{max}}$ and $\Delta K_{wip_{max}}$ were set to 7 and 10.
- the maximum number of iterations was set to 10.
- the convergence threshold θ was set to 10^{-4}.

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

3.1. Evaluation on RM1

3.1.1. Corpus and Acoustic Modeling

The 3,990 speaker-independent (SI) training utterances from 109 speakers were used for model training. Evaluation was performed on the 300 utterances from 10 speakers in the SI Feb'91 test set using the standard word-pair grammar of perplexity 60. The speaker-dependent (SD) development data set, consisting of 1,200 utterances from 12 speakers was used for the estimation of the decoding parameters.

Forty-seven context-independent phoneme models were trained using the SI training set. There are 10 Gaussian mixtures per state in each phoneme CDHMM.

3.1.2. Experimental Results

The decoding parameters were optimized using the RM1 SD development data set and the iterative LP algorithm described in Algorithm 1. The convergence of the grammar factor and word insertion penalty are shown in Fig. 1 and Fig. 2 respectively, whereas the corresponding recognition performance on the test data is shown in Fig. 3. It is clear that the algorithm converges in all of the four different initial settings in 5–7 iterations to the *same* optimal values: $(K_{gf}, K_{wip}) = (3.5, 5.23)$. The corresponding word and utterance recognition accuracies are 93.44% and 71.67% respectively, which are better than the ones computed with the decoding parameters found by an extensive grid search.

3.2. Evaluation on WSJ0

3.2.1. WSJ0 Corpus and Acoustic Modeling

The standard SI-84 training set was used for training the 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 CDHMM has a Gaussian mixture density of 16 components per state, and there are totally 3,132 tied states. "Optimal" decoding parameters were found by an extensive grid search using the test set and the si_dt_05 development set[2]. The SI model has a word and an utterance recognition accuracy of 93.16% (92.92%) and 44.55% (44.55%) respectively using the decoding parameters found by a grid search over the test (development) data.

3.2.2. Experimental Results

Our iterative LP algorithm was again run with the 4 different initial values as it was done in the RM experiment on the 442-utterance subset of the si_dt_05 development set to determine the decoding parameters. The algorithm once again converged in 5–7 iterations to give the same optimal decoding parameters $(K_{gf}, K_{wip}) = (12.6, -6.58)$. The ensuing word and utterance accuracy are 92.53% and 42.42% respectively which are slightly worse than but comparable with the results obtained with the decoding parameters found by a grid search on the development data.

[2] Only 442 from 10 speakers out of the total 1,206 utterances in si_dt_05 were employed because not all words in the remaining utterances are covered by the WSJ0 bigram language model.

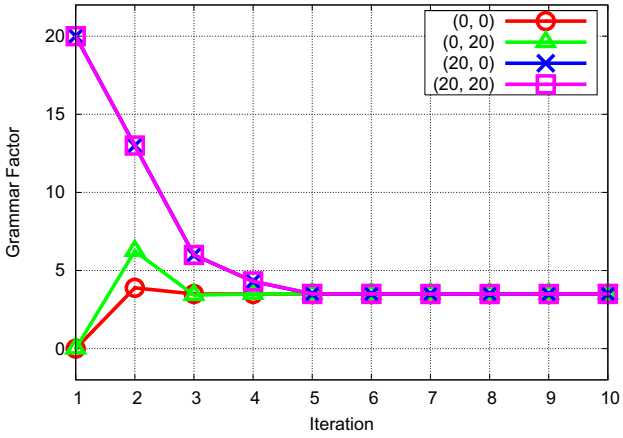

Figure 1: Iterative LP optimization of the grammar factor on RM1 using various initial (K_{gf}, K_{wip}) values.

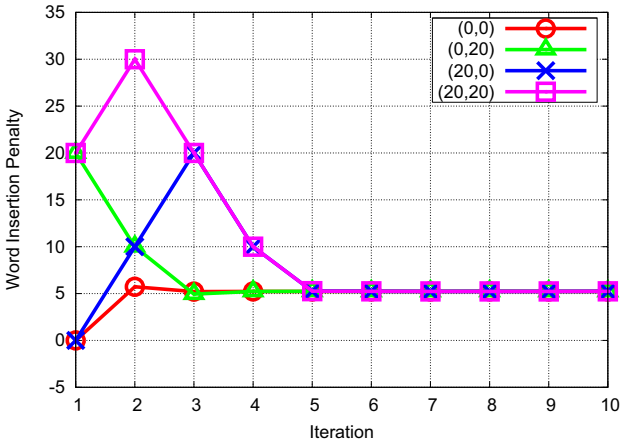

Figure 2: Iterative LP optimization of the word insertion penalty on RM1 using various initial (K_{gf}, K_{wip}) values.

We compute the perplexities of the 442-utterance subset of the si_dt_05 development set and the test set, and the results are 121 and 111 respectively. We hypothesize that there may be some mismatch in the statistics of words in the two data sets. Thus, we further performed the following experiment: we divided the test set equally into 2 subsets A and B with no overlapping speakers; their perplexities are 113 and 108 respectively. Decoding parameters were estimated from test subset A and they were then used to decode test utterances in test subset B, and vice versa. The combined word and utterance accuracies are now 92.88% and 43.64% respectively; the findings are summarized in Table 2.

The experiments show that if the word statistics in the data used to train the decoding parameters matches well with that in the test data, the proposed iterative LP algorithm can effectively find a good set of values for the decoding parameters.

4. Conclusions

We investigate a discriminative and iterative linear programming (LP) algorithm to estimate the (locally) optimal values of the decoding parameters for ASR. The LP solution at each iteration is globally optimal for the particular LP problem in the iteration. Taking all the iterations together, the algorithm will give a locally (if not globally) optimal solution. Empirically we observe that the algorithm is effective: the performance of speech recognition using the estimated decoding parameters is comparable to that using decoding parameters found by an extensive grid search. Moreover, the algorithm converges quickly within 5–7 iterations, and the results seem to be independent of the initial values used to run the algorithm.

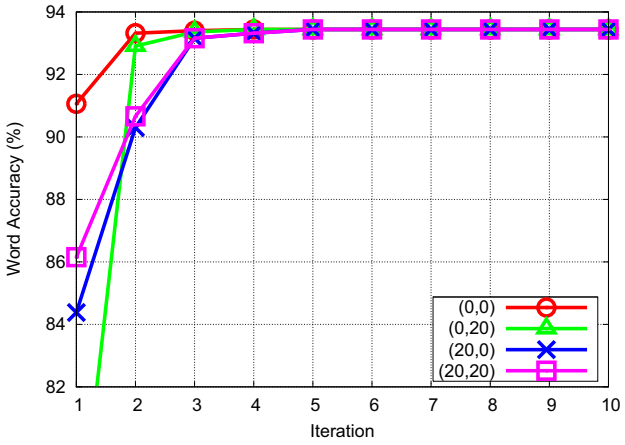

Figure 3: RM1 word accuracy using the decoding parameters found by iterative LP with various initial (K_{gf}, K_{wip}) values.

Table 2: WSJ0 recognition performance using the decoding parameters estimated from various data sets. The numbers in parentheses are the number of utterances used in the data set.

Training Set	Method	Word (Utt.) Accuracy	K_{gf}	K_{wip}
test set (330)	grid search	93.16% (44.55%)	15	-30
dev set (330)	grid search	92.92% (44.55%)	14	-10
dev set (442)	iterative LP	92.53% (42.42%)	12.6	-6.58
2-fold test set	iterative LP	92.88% (43.64%)	12.8	-26.3
(164, 166)			13.95	-16.9

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.

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

[3] Tadashi Emori, Yoshifumi Onishi, 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.