SUMMARY This paper describes a technique for overcoming the model shrinkage problem in automatic speech recognition (ASR), which allows application developers and users to control the model size with less degradation of accuracy. Recently, models for ASR systems tend to be large and this can constitute a bottleneck for developers and users without special knowledge of ASR with respect to introducing the ASR function. Specifically, discriminative language models (DLMs) are usually designed in a high-dimensional parameter space, although DLMs have gained increasing attention as an approach for improving recognition accuracy. Our proposed method can be applied to linear models including DLMs, in which the score of an input sample is given by the inner product of its features and the model parameters, but our proposed method can shrink models in an easy computation by obtaining simple statistics, which are square sums of feature values appearing in a data set. Our experimental results show that our proposed method can shrink a DLM with little degradation in accuracy and perform properly whether or not the data for obtaining the statistics are the same as the data for training the model. 

key words: model shrinkage problem, discriminative language model, linear model, feature selection

1. Introduction

Automatic speech recognition (ASR) technology, which faithfully transcribes speech signals into word sequences, plays an important role in extracting linguistic information from human utterances. ASR has been practically employed in many applications including speech translation and spoken document retrieval [1]–[9], and its applicability will extend to various fields.

Certain issues must be dealt with in terms of encouraging the widespread use of ASR. Specifically, training an acoustic model (AM) and a language model (LM) is difficult for application developers and users without special knowledge of ASR. Model training requires a large amount of data, which might need reference labels, and there are many hyperparameters for training. Furthermore, some recent model training methods assume the use of a large-scale parallel computation system.

One way to solve this problem is for ASR specialists to provide general purpose models for developers and users. Recently, discriminative training has succeeded in increasing the generalization ability of models and this has been encouraged by the recent availability of large-scale corpora. However, this tends to generate a very large model, and therefore a large process memory is required.

For example, increasing attention is being paid to discriminative language models (DLMs) in the ASR research area [10]–[16]. A DLM is generally designed as a high-dimensional model and is usually larger than standard n-gram language models that are estimated based on the maximum likelihood criterion. Furthermore, DLMs assume the simultaneous use of a standard LM.

A long length of n-grams, i.e. 3 or 4-grams, plays an important role in providing an accurate result. In addition, since the long length of n-grams is sparsely distributed, we have to prepare a large number of entries to cover a variety of domains. If we omitted some of the entries without careful consideration, recognition errors would increase greatly. When building a general purpose LM it is natural for the LM to be large.

In this paper, we describe the model shrinkage problem (MSP). This is to provide a framework for controlling model size. For example, the default model is used when a large memory is available. If there is a limitation, the model is shrunk to one that is compact and performs with less or little degradation of accuracy. The purpose of this suggestion is to generate a technique to provide application developers and users with freedom of choice in terms of model size.

Model shrinkage should be employed for all ASR models. This paper deals with DLMs as the first step, but our proposed method can be applied to linear models, in which the score of an input sample is given by the inner product of its features and the model parameters. Our proposed method shrinks a DLM based on its parameters and simple statistics, which are square sums of the feature values that appear in a data set. The statistics can be obtained from a data set without reference labels, i.e. ASR hypothesis word sequences. The data set can be different from that used for training the DLM. Therefore, the DLM can be shrunk simply by obtaining the statistics from the ASR results in a usage environment, even if the DLM did not originally include the statistics for the model shrinkage.

The MSP is different from the feature selection problem [17]. In the MSP, the shrinkage model must be generated as soon as a user specifies the model size. In addition, the already estimated model parameters can be used in the MSP. However, some methods for dealing with the feature selection problem can be applied to the MSP. This is also discussed in the paper. Other related work includes that on LM pruning, in which some n-grams are removed and the
other n-grams have their probabilities and back-off weights modified [18]–[20]. Some pruning methods can also be applied to the MSP directly or with some modifications. These methods are developed only for backoff n-gram LMs. Our proposed method is designed for linear models including DLMs.

This paper is organized as follows: we begin by describing DLMs in Sect. 2. In Sect. 3, we describe the MSP and our proposed method. Section 4 provides our experimental results. And Sect. 5 concludes this paper.

2. Discriminative Language Models Based on Linear Model

2.1 Reranking with a Linear Model

Given a j-th word sequence of multiple ASR hypotheses for the i-th utterance, we represent the feature vector of the word sequence as \( f_{i,j} \). Features for DLMs are typically designed as a set of n-gram counts appearing in a word sequence. Each element of the feature vector corresponds to the occurrence count of a certain n-gram. Namely, where the k-th element of the feature vector \( f_{i,j} \) is represented as \( f_{i,j,k} \),

\[
    f_{i,j,k} = \text{the count of the k-th n-gram in the j-th word sequence for the i-th utterance.}
\]

For example, if a bigram ‘yeah I’ corresponds to the k-th element of the vector, \( f_{i,j,k} \) is the count of ‘yeah I’ appearing in the word sequence.

Given an n-best list of i-th utterance \( L_i = \{ f_{i,j} | j = 1, 2, \cdots, N_i \} \) and DLM parameters \( a \), the goal of speech recognition is to find the hypothesis from \( L_i \) that maximizes the following equation,

\[
    a_0 f_{i,j,0} + a^\top f_{i,j}.
\]

where \( f_{i,j,0} \) is a log scale recognition likelihood given by an AM and a standard LM. \( a_0 \) is a scaling constant and is decided using a development set. \( a^\top \) denotes the transpose of the matrix.

2.2 Reranking Model Training

For training, we require a data set that comprises:

- N-best lists \( \{ L_i | i = 1, 2, \cdots, I \} \)
  They are generated from a speech recognizer for training speech data that consist of I utterances.
- Word error rate (WER) as sample weight
  The WER of each word sequence hypothesis is used for sample weight training, where a sample is a word sequence in the lists.

The parameters are estimated by finding a parameter vector \( a \) that minimizes a predefined objective function. In this paper, we use the round-robin duel discrimination (R2D2) model [21], whose objective function is given as follows,

\[
    O_{R2D2} = \sum_{i=1}^{I} \log \sum_{j=1}^{N_i} \sum_{j'=1}^{N_i} \frac{\exp(\sigma_1 e_{i,j}) \exp(a^\top f_{i,j})}{\exp(\sigma_2 e_{i,j'}) \exp(a^\top f_{i,j'})}.
\]

Here \( \sigma_1 \) and \( \sigma_2 \) are hyperparameters that should be decided using a development set. R2D2 provides an accurate model that is robust for difference of tasks and converges with the global optimum because of the concavity of the objective function. Although the double summation as regards \( j \) and \( j' \) seems to be computationally expensive, there is an efficient calculation method [21].

To estimate an optimal solution, we can alternatively use the gradient descent method or the quasi-Newton method. In practice, the minimization problem with \( a \) is modified by introducing a regularization term to mitigate the overfitting risk. We employ L2-norm for our experiments, that is, the minimized function is given as

\[
    O_{R2D2} + C ||a||^2.
\]

\( C \) is a constant, which should be decided by using a development set.

3. Model Shrinkage

We first provide an overview of the MSP in this section. Next we present the principle of our proposed method in Sects. 3.2 to 3.6. Finally, we summarize our proposed method by describing the steps of the process in Sect. 3.7.

3.1 Model Shrinkage Problem

Given an original DLM parameter vector \( a \), the MSP is a problem related to constructing a function \( f(ma) \) that converts \( a \) to a shrinkage model \( \hat{a}_m \) for any \( m \), in which \( \hat{a}_m \) provides the result most similar to \( a \). \( m \) denotes the number of dimensions, i.e. the model size.

Even if \( f \) is a function that simply selects a subset of \( a \), the total number of subsets is \( \sum_{m=1}^{n} \binom{n}{m} \) where \( n \) is the size of \( a \). If \( f \) is a function that converts values of \( a \)’s elements, the problem becomes more difficult. It is impossible to obtain the complete solution to the MSP in a polynomial time, if we use a brute-force search.

In addition, a shrunk model \( \hat{a}_m \) should be generated without loading the whole of \( a \), because the computer employed in a normal usage environment might not sufficient memory to hold the default size of model. We have to read a small size model directly from a hard disc drive, etc.

3.2 Practicable Formulation of Model Shrinkage Problem

Assume we have a discrimination function \( \phi(f | a) \), i.e. the linear model in this work, where \( a = (a_1, a_2, \cdots, a_n)^\top \in \mathbb{R}^n \) is a vector of the model parameters and \( f \) is an n-dimensional feature vector. Let a data set \( S = \{ f_i | i = 1, 2, \cdots, N \} \) where \( f_i \) is a feature vector, \( f_i \) corresponds to \( f_{i,j} \) for DLMs. Thus \( N = \sum_{i=1}^{I} N_i \). In addition, we assume a function family \( \phi_m \) for \( 1 \leq m \leq n \). Given a shrunk model size \( m \), we
regard the MSP as a problem that involves finding a parameter set \( \hat{\mathbf{a}}_m = (\hat{a}_1, \cdots, \hat{a}_m) \)^T \in \mathbb{R}^m \) that satisfies the following conditions.

**Condition 1:** Modified parameters \( \hat{\mathbf{a}}_m = (\hat{a}_1, \cdots, \hat{a}_m) \)^T is an arbitrary function, i.e., \( \phi_m(f|\mathbf{a}) \) is as similar to \( \phi(f|\mathbf{a}) \) as possible, i.e.

\[
\min_{f \in \mathcal{S}} \sum_{n \in \mathcal{S}} (\phi(f|\mathbf{a}) - \phi_m(f|\hat{\mathbf{a}}_m))^2
\]

\[
\sum_{a \in \mathcal{S}} (\phi(f|\mathbf{a}) - \phi_m(f|\hat{\mathbf{a}}_m))^2 = \min_{a \in \mathcal{S}} (\phi(f|\mathbf{a}) - \phi_m(f|\hat{\mathbf{a}}_m))^2
\]

(4)

Condition 1 requires that \( \hat{\mathbf{a}}_m \) can be immediately obtained for any \( m \) simply by choosing the first \( m \) dimensions of \( \hat{\mathbf{a}}_n \), and Condition 2 requires \( \hat{\mathbf{a}}_m \) to be the best among all of the \( m \)-dimensional parameters.

Note that \( \phi_m \) is a pre-defined function and only \( \hat{\mathbf{a}}_m \) is estimated. Moreover, \( \hat{\mathbf{a}}_n \) has \( n \) elements, but the elements are permuted from \( \mathbf{a} \). \( \hat{\mathbf{a}}_k \) generally does not correspond to \( a_k \).

To distinguish the conversion of parameter values and the permutation, we assume that \( \mathbf{a} \) is converted to \( \hat{\mathbf{a}}_m \) with two matrices, \( B \) and \( R_n \), where \( R_n = (r_1, \cdots, r_n) \) and \( r_k \in \mathbb{R}^n \), i.e.,

\[
\hat{\mathbf{a}}_n = R_n^\top B \mathbf{a}.
\]

(5)

\( B \) is an \( n \times n \) matrix for converting parameter values and \( R_n \) is an arbitrary \( n \)-dimensional orthogonal basis, i.e., \( r_k^\top r_k = \delta_{kk'} \) (\( \delta_{kk'} \) is Kronecker delta) for permutating the elements. \( r_k \) is sparse which means that only one element of \( r_k \) has a value of 1 and all of the other elements are zero. \( R_n \hat{\mathbf{a}}_n \) returns the permutation of the elements of \( \hat{\mathbf{a}}_n \) to that of \( \mathbf{a} \). \( \hat{\mathbf{a}}_m \) can be mapped to an \( n \)-dimensional vector by \( R_m \hat{\mathbf{a}}_m \), in which the permutation of the elements corresponds to that of \( \mathbf{a} \).

**3.3 Model Shrinkage of Linear Discrimination Function**

In this work, we assume a linear function \( \phi \) as

\[
\phi(f|\mathbf{a}) = \mathbf{a}^\top f
\]

and \( \phi_m \) as

\[
\phi_m(f|\hat{\mathbf{a}}_m) = \hat{\mathbf{a}}_m^\top R_m^\top f.
\]

(7)

This formula means that \( R_m \hat{\mathbf{a}}_m \) is an approximation of \( \mathbf{a} \), i.e.

\[
R_m \hat{\mathbf{a}}_m \approx \mathbf{a}
\]

and thus,

\[
\hat{\mathbf{a}}_m \approx R_m^\top \mathbf{a}.
\]

(9)

In this case, \( R_m^\top \) is a transform for choosing \( m \) dimensions out of \( n \) dimensions.

**3.4 Estimation of Weights**

For simplicity, we assume that \( B \) is a diagonal matrix:

\[
B = \begin{pmatrix}
B_{11} & 0 & \cdots & 0 \\
0 & B_{22} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & B_{nn}
\end{pmatrix}
\]

(10)

The weight \( b_k \) is interpreted as the importance of \( k \)-th dimension and can be an indicator for determining \( R_n \).

Let us consider the case of \( m = n \). \( \phi_n \) can be written as

\[
\phi_n(f|\hat{\mathbf{a}}_n) = \hat{\mathbf{a}}_n^\top R_n^\top f = \mathbf{a}B_R R_n^\top f = \mathbf{a}^\top B f
\]

(11)

It is obvious that \( B = I_n \) is the optimum solution with respect to Condition 2, where \( I_n \) denotes an \( n \times n \) identity matrix. This means that we lose the indicators for determining \( R_n \).

To avoid this situation, we re-formulate Eq. (4) under two assumptions. \( b_k \)s can be obtained as a solution of the re-formulated equation. And \( R_n \) is decided using the \( b_k \)s.

**Assumption 1:** The difference between the original discrimination function \( \phi(f|\mathbf{a}) \) and the “to be shrunk” function \( \phi_n(f|\hat{\mathbf{a}}_n) \) can be assessed dimension by dimension, i.e.,

\[
e_n(B) = \sum_{f \in \mathcal{S}} \|F \mathbf{a} - F B \mathbf{a}\|^2 \approx \sum_{f \in \mathcal{S}} (\mathbf{a}^\top f - \mathbf{a}^\top B f)^2
\]

(12)

where \( F \) denotes an \( n \times n \) diagonal matrix whose diagonal elements are \( f \).

**Assumption 2:** The diagonal elements of \( B \) are sparse, i.e., a few number of \( b_k \) values should be large (nearly one) and the others should be nearly zero.

Let us explain the first assumption. Letting \( \mathbf{f} = (f_1, f_2, \cdots, f_n)^\top \), the right hand side of Eq. (12) is

\[
\sum_{f \in \mathcal{S}} (\mathbf{a}^\top f - \mathbf{a}^\top B f)^2 = \sum_{k=1}^{n} \left( \sum_{k=1}^{n} a_k f_k - \sum_{k=1}^{n} a_k b_k f_k \right)^2
\]

(13)

and the left hand side is

\[
\sum_{f \in \mathcal{S}} \|F \mathbf{a} - F B \mathbf{a}\|^2 = \sum_{k=1}^{n} \sum_{k'=1}^{n} (a_k f_k - a_k b_k f_k)^2
\]

(14)

\[
= \sum_{k=1}^{n} \sum_{k'=1}^{n} d(k)^2
\]

(15)

where we represent \( a_k f_k - a_k b_k f_k \) as \( d(k) \). Then Eq. (13) can be written as

\[
\sum_{f \in \mathcal{S}} (\sum_{k=1}^{n} a_k f_k - \sum_{k=1}^{n} a_k b_k f_k)^2 = \sum_{f \in \mathcal{S}} (\sum_{k=1}^{n} d(k))^2
\]

(16)

\[
= \sum_{f \in \mathcal{S}} \sum_{k=1}^{n} d(k)^2 + \sum_{f \in \mathcal{S}} \sum_{k=1}^{n} \sum_{k' \neq k} d(k)d(k')
\]

(17)

The first term of Eq. (17) is identical to Eq. (15), and the
second term can be written as
\[\sum_{i \in S} \sum_{k = 1}^{n} \sum_{l \in S} a_i a_{k l} (1 - b_k) (1 - b_k) f_i f_k.\] (18)
Here, if \(f\) is sufficiently sparse, most of \(f_i f_k\)'s have zero. Therefore, we can regard Eq. (12) as a proper assumption when \(f\) is sparse.

The assumption 2 assumes that only a small number of \(b_k\)'s should have large (nearly one) values and others should have values nearly zero. As mentioned above, the optimum condition for \(b_k\) is \(b_k = 1\) for all \(k\). However, if an impact of dimension \(k\) over the value of \(\phi_n\) is small, \(b_k\) could be smaller than 1.

According to the above two assumptions, we define an objective function as
\[\tilde{\varepsilon}_n(B) = \sum_{i \in S} \|F a - F B a\|^2 + c \|b\|_1.\] (19)
where \(b = (b_1, b_2, \cdots, b_n)^T\) and \(\|\cdot\|_1\) denotes the L1-norm. The second term of the right hand of Eq. (19) is a regularization term and \(c\) is a constant for making \(b\) sparse, where \(c \geq 0\). \(c\) is decided after specifying the shrinkage model size \(m\). This is described in Sect. 3.6.

Equation (19) can be written as
\[\tilde{\varepsilon}_n(B) = \sum_{i \in S} \sum_{k = 1}^{n} (a_i f_k - a_i b_k f_k)^2 + \sum_{k = 1}^{n} c b_k\] (20)
\[= \sum_{k = 1}^{n} \left((1 - b_k)^2 a_k^2 \sum_{i \in S} f_i^2 + c b_k\right).\] (21)
Assuming \(b_k \geq 0\),
\[\tilde{\varepsilon}_n(B) = \sum_{k = 1}^{n} \left((1 - b_k)^2 a_k^2 \sum_{i \in S} f_i^2 + c b_k\right).\] (22)
The minimization problem of \(\tilde{\varepsilon}_n(B)\) can be considered separately with respect to each element. Thus we represent it as
\[\tilde{\varepsilon}_{n,k}(B) = (1 - b_k)^2 a_k^2 \sum_{i \in S} f_i^2 + c b_k\] (23)
\[= \left(a_k^2 \sum_{i \in S} f_i^2\right) \left(b_k - (1 - \frac{c}{2 a_k^2 \sum_{i \in S} f_i^2})\right)^2 + a_k^2 \sum_{i \in S} f_i^2 - (1 - \frac{c}{2 a_k^2 \sum_{i \in S} f_i^2}) a_k^2 \sum_{i \in S} f_i^2.\] (24)
Since \(\tilde{\varepsilon}_{n,k}(B)\) is a quadratic function with respect to \(b_k\) where \(a_k^2 \sum_{i \in S} f_i^2 \neq 0\), the solution is
\[b_k = 1 - \frac{c}{2 a_k^2 \sum_{i \in S} f_i^2},\] (25)
if \(c \leq 2 a_k^2 \sum_{i \in S} f_i^2\) and \(b_k = 0\) otherwise. If \(a_k^2 \sum_{i \in S} f_i^2 = 0\), the solution is \(b_k = 0\) because \(\tilde{\varepsilon}_{n,k}(B) = c b_k\).

As \(b_k < 0\), there is no solution where \(c \geq 0\). Therefore,
\[b_k = \max \left(0, 1 - \frac{c}{2 a_k^2 \sum_{i \in S} f_i^2}\right)\] (26)
It is valuable to mention the range of the \(b_k\) value for the later discussion. \(0 \leq b_k \leq 1\) because \(c \geq 0\).

### 3.5 Determination of Sorting Matrix \(R_n\)

The transform \(B\) can be determined by Eq. (26). Then how do we determine \(R_n\)? For this purpose, let us consider the case when \(m = n - 1\).
\[\tilde{a}_{n-1} = R_n^* B a\] In this case,
\[c_{n-1}(B) = \sum_{i \in S} \|F a - F R_{n-1} R_n^* B a\|^2.\] (27)

Here,
\[R_{n-1} R_{n-1}^T = (r_1, r_2, \cdots, r_{n-1}) (r_1, r_2, \cdots, r_{n-1})^T\] (28)
\[= \sum_{k = 1}^{n-1} r_k^T r_k = I_n - r_n r_n^T = I_n - Z_n\] (29)
where \(Z_n = r_n r_n^T\), i.e. \(Z_n\) is an \(n \times n\) diagonal matrix whose diagonal elements are \(r_n\). When the \(K\)-th element of \(r_n\) is one,
\[c_{n-1}(B) = \sum_{i \in S} \|F a - F B a + F Z_n B a\|^2\] (30)
\[= \sum_{i \in S} \sum_{k = 1}^{n} (f_i a_k - f_i b_k a_k + f_k b_k a_k)^2\] (31)
\[= \sum_{i \in S} \left(\sum_{k \neq K} (f_i a_k - f_i b_k a_k)^2 + (f_k a_k)^2\right)\] (32)
\[= \sum_{i \in S} \left(\sum_{k = 1}^{n} d_k(f)^2 + a_k^2 b_k^2 (2 - b_k)\right)\] (33)
\[= c_{n-1}(B) + a_k^2 b_k (2 - b_k) \sum_{i \in S} f_i^2\] (34)
\[= c_{n-1}(B) + a_k^2 b_k (2 - b_k) \sum_{i \in S} f_i^2\] (35)

Here we write
\[\eta_k(B) = a_k^2 b_k (2 - b_k) \sum_{i \in S} f_i^2.\] (36)
where \(\eta_k(B)\) means the impact of removing the \(K\)-th dimension on the total error. As \(0 \leq b_k \leq 1\), \(\eta_k(B)\) decreases monotonically with \(b_k\). To satisfy Condition 1, we have to choose \(K\) (and therefore \(r_n\)) so that \(\eta_k(B)\) is minimized, i.e.,
\[K_n = \arg \min_k \eta_k(B).\] (37)
We can calculate sequence \(K_1, K_2, \cdots, K_n\), which is a permutation of \(1, 2, \cdots, n\) that satisfies
\[\eta_{K_1}(B) \geq \eta_{K_2}(B) \geq \cdots \geq \eta_{K_n}(B).\] (38)
Then we can determine a basis \(r_k\) so that the \(K_i\)-th element is one and the other elements are zero.

Considering Eq. (25), therefore where \(c \leq 2 a_k^2 \sum_{i \in S} f_i^2\) and \(a_k^2 \sum_{i \in S} f_i^2 \neq 0\), \(\eta_k(B)\) can be calculated as
\[
\eta_K(B) = a_k^2 b_K (2 - b_K) \sum_{i \in S} f_k^2
\] (39)

\[
= \left(a_k^2 \sum_{i \in S} f_k^2 \right) \left(1 - \frac{c}{2a_k^2 \sum_{i \in S} f_k^2} \right)
\times \left(1 + \frac{c}{2a_k^2 \sum_{i \in S} f_k^2} \right)
\] (40)

\[
= \left(a_k^2 \sum_{i \in S} f_k^2 \right) \left(1 - \frac{c^2}{4a_k^2 \sum_{i \in S} f_k^2} \right) (41)
\]

\[
= a_k^2 \sum_{i \in S} f_k^2 - \frac{c^2}{4a_k^2 \sum_{i \in S} f_k^2}. (42)
\]

Hence,
\[
K_n = \arg \min_k \eta_K(B)
\] (43)

\[
= \arg \min_k \left( a_k^2 \sum_{i \in S} f_k^2 + \frac{-c^2}{4a_k^2 \sum_{i \in S} f_k^2} \right), (44)
\]

since the both terms are monotonic increasing functions with \(a_k^2 \sum_{i \in S} f_k^2 > 0)\),
\[
K_n = \arg \min_k a_k^2 \sum_{i \in S} f_k^2. (45)
\]

Considering Eq. (36), when \(b_K = 0\), \(\eta_K(B) = 0\) and this is the minimum bound of \(\eta_K(B)\), because \(0 \leq b_K \leq 1\). As mentioned above, \(b_K = 0\) if \(a_k^2 \sum_{i \in S} f_k^2 = 0\). This means that Eq. (45) can also be applied when \(a_k^2 \sum_{i \in S} f_k^2 = 0\), although this case is not defined in Eq. (44).

3.6 Value for Sparseness Constant \(c\)

\(c\) controls the sparseness of \(b_K\). When \(b_K\) is zero, it generates model parameters of zero, i.e. \(b_K a_k\). Thus, \(\hat{a}_m\) might include zero parameters. To avoid this, we decide the \(c\) value so that the top \(m\) elements of \(\hat{a}_m\) have non-zero values and the others are zero. Considering Eq. (26),
\[
c = 2a_{K+1}^2 \sum_{i \in S} f_{K+1}^2
\] (46)

where \(a_{K+1}\) and \(f_{K+1}\) are defined as zero.

3.7 Summary of Our Proposed Method

We describe the flow of our proposed method assuming the use on DLMs.

**Preparation:**

1. Calculate \(q_k = a_k^2 \sum_{i=1}^{I} \sum_{j=1}^{N_i} f_{i,j,k}^2\) for all \(k = 1, 2, \ldots, n\).
2. Sort \(q_k\) in decreasing order.

This process corresponds to constructing \(R_n\). In short, the top of the sorted vector corresponds to \(K_1\) and the bottom corresponds to \(K_n\).

**Generating \(\hat{a}_m\):**

1. Set \(c = 2q_{K+1}\).
2. Generate \(\hat{a}_m = (b_K, a_K, b_{K,1}, \ldots, b_{K,n}, a_{K,n})^T\) where \(b_K\) is given as Eq. (26).

We also propose a special case method of the proposed method, in which \(\hat{a}_m\) consists of \(m\) elements of \(a\), \((a_K, a_{K,1}, \ldots, a_{K,n})^T\). This can be regarded as our original proposed method under the constraint that \(b_K = 0\) or \(1\). In this case, we do not need to memorize \(q_k\) values.

4. Experiments

4.1 Comparison of Methods for Obtaining Shrinking Models

We introduce three other methods for solving the MSP. One is the simplest method and the others originate from the feature selection problem. Methods for the feature selection problem are generally categorized into two approaches: the filter approach and the wrapper approach. We selected one method from each approach and applied it to the MSP. We describe these three methods and compare them with our proposed methods.

- **Power based cut-off**

\(\hat{a}_m\) is constructed from the \(a_k\)'s with the largest \(|a_k|\). In other words, the permutation is decided in decreasing order of \(|a_k|\).

- **Filter approach with mutual information criterion**

We employ a method that is a type of filter approach and decides the permutation of the parameters based on the mutual information (MI) criterion [22]–[24].

We regard an oracle word sequence, which is a sample with the minimum WER in an n-best list, as a reference of the list. If there are multiple word sequences with the minimum WER, we select one of them at random. We represent the index of the reference for the \(i\)-th list as \(r_i(= \arg \min_{j \in [1, 2, \ldots, N]} e_{i,j})\). And, where \(x \in [0, 1]\) and \(y \in [0, 1]\), we define \(M_k(x, y)\) as

\[
M_k(0, 0) = \sum_{i=1}^{I} \sum_{j=1}^{N_i} \delta_{0,f_{i,j,k}} (1 - \delta_{j,r_i})
\] (47)

\[
M_k(1, 0) = \sum_{i=1}^{I} \sum_{j=1}^{N_i} (1 - \delta_{0,f_{i,j,k}}) (1 - \delta_{j,r_i})
\] (48)

\[
M_k(0, 1) = \sum_{i=1}^{I} \sum_{j=1}^{N_i} \delta_{0,f_{i,j,k}} \delta_{j,r_i}
\] (49)

\[
M_k(1, 1) = \sum_{i=1}^{I} \sum_{j=1}^{N_i} (1 - \delta_{0,f_{i,j,k}}) \delta_{j,r_i}
\] (50)

\(M_k\) denotes the number of samples, in which \(x = 1\) means we count samples satisfying \(f_{i,j,k} \neq 0\) and \(y = 1\) means we count the oracle samples. Zero values for \(x\) and \(y\) mean we count samples that do not satisfy the the respective cases.

To obtain the MI value, we calculate \(P_k(x, y) = \frac{M_k(x,y)}{N}\) using a data set, where \(N = \sum_{i=1}^{I} N_i\). Here we define
representations \( P_k(x) = P_k(x, 0) + P_k(x, 1) \) and \( P_k(y) = P_k(0, y) + P_k(1, y) \), an the MI value is given as

\[
I_k = \sum_{x \in \{0, 1\}} \sum_{y \in \{0, 1\}} P_k(x, y) \log \frac{P_k(x, y)}{P_k(x)P_k(y)}.
\]

The permutation of the parameters is defined as the decreasing order of the \( I_k \) values.

**Wrapper approach using SBS**

We employ a type of wrapper approach for the feature selection problem, namely the sequential backward selection (SBS) method [25]. We use the minimum square error as a criterion for SBS,

\[
\sum_{i,j} (a^\top f_{i,j} - a^\top \beta f_{i,j})^2 \tag{52}
\]

where \( b_k \) is 0 or 1.

SBS first initializes \( b \), in which all the elements are set at 1. Next, when converting an element of \( b \) to 0, SBS finds an element that minimizes Eq. (52) and memorizes this element index. And \( b \) is updated, by which the value of the memorized element is set at 0. SBS again finds an element that minimizes Eq. (52) from the remaining elements. SBS continues these processes until \( b \) becomes a zero vector. The inverse order of the selected elements corresponds to exactly the permutation for creating \( \hat{a}_m \).

It is interesting to note that SBS based on Eq. (52) is equivalent to the sequential forward selection (SFS) based on

\[
\sum_{i,j} (a^\top \beta f_{i,j})^2. \tag{53}
\]

The purpose of the feature selection problem is to find the optimal subset of the parameters. Therefore, a stopping criterion is typically defined when employing SBS or SFS for the feature selection problem. This is effective specifically when the number of parameters is very large. However, the loop of SBS and SFS should be continued until all the parameters are selected during application to the MSP, because the purpose is to decide the priority of the parameters.

Table 1 summarizes the characteristics of these methods and our proposed methods. ‘Use of model’ denotes that the original model \( a \) is required to decide the permutation of the parameters. ‘Use of data’ denotes that data are needed to decide the permutation. Specifically, MI requires data with reference labels, while our proposed methods and SBS do not need labels. The computation cost of SBS is much higher than that of the others. The power based cut-off only sorts the parameters. Moreover, SBS needs to access all the data iteratively, while our proposed methods and the MI approach access the data only once.

### 4.2 Evaluations of Shrinkage Approaches

The Corpus of Spontaneous Japanese (CSJ) [26] includes many lectures given in Japanese and their transcriptions. The lecture data can be divided into academic lectures and others on various themes such ‘My hometown’ and ‘The happiest thing in my life’. We call the latter lectures ‘omnibus’ for convenience in this paper.

Table 2 shows the amount of data used for obtaining a DLM and for evaluating the model shrinkage methods. Sets A-Large, A-Small and A-Eval consist of academic lectures. Sets O-Small and O-Eval consist of omnibus lectures.

We generated 5000-best lists of all the utterances shown in Table 2. The speech recognizer we used is a weighted finite state transducer based decoder, SOLON, which was developed at NTT Communication Science Laboratories. SOLON can provide a fast efficient search by using a fast on-the-fly composition algorithm [27]. We employed a Kneser-Ney smoothed word n-gram model [28], which was trained using all the other academic data not shown in Table 2 to simulate a test-set environment for DLM training [29]. We call this a standard LM hereafter. The lexicon size is 51,474. The AM is Gaussian mixed tri-phone HMMs, which were trained using academic lectures consisting of set A-Large and training data for the standard LM. We employed discriminative training with a minimum classification error criterion for parameter estimation [30].

Using set A-Large, we trained several DLMs based on Eq. (3) using different values for hyperparameters \( \sigma_1 \), \( \sigma_2 \) and \( C \). The features were counts of the word unigrams, bigrams and trigrams and POS unigrams, bigrams and trigrams. Set A-Small was used to obtain the best model and the optimal value of the scaling constant \( a_0 \).

Table 4 shows the parameter size of the LMs, the standard (Kneser-Ney smoothed) LM and the DLM. The size of the DLM means the number of parameters with a non zero value. The DLM is much bigger than the standard LM. Specifically, trigram features occupy about 80%.

Table 5 shows WER before and after applying the DLM to the 5000-best lists. If the absolute differences of the WERs are 0.2 and 0.3 for sets A-Eval and O-Eval, respectively, they are statistically significant (\( p < 0.02 \)).

We evaluated the model shrinkage methods under the four conditions shown in Table 3. To obtain information

### Table 1 Characteristics of methods for the MSP.

| Method      | Use of model | Power  | MI      | SBS  |
|-------------|--------------|--------|---------|------|
| Proposed    | ✓            | ✓      | -       | ✓    |
| methods     |              |        |         |      |

### Table 2 Data for DLM training and MSP evaluation in CSJ.

|        | # of uttr. | # of words |
|--------|------------|------------|
| A-Large| 25,130     | 420,918    |
| A-Small| 1,293      | 26,329     |
| A-Eval | 1,156      | 26,798     |
| O-Small| 1,096      | 19,198     |
| O-Eval | 1,095      | 19,034     |
Table 3 Evaluation conditions for the MSP.

|          | DLM training | DLM development | MSP training | Evaluation | Remarks                  |
|----------|--------------|-----------------|--------------|------------|--------------------------|
| Eval-Small | A-Large      | A-Small         | A-Small      | A-Eval     | Small training data      |
| Eval-Large | A-Large      | A-Small         | A-Large      | A-Eval     | Large training data      |
| Eval-Cross | A-Large      | A-Small         | A-Large      | O-Eval     | Small training data      |
|           |              |                 |              |            | Cross-domain evaluation  |

Table 4 Number of parameters of standard LM and DLM in CSJ.

|          | 1-grams | 2-grams | 3-grams |
|----------|---------|---------|---------|
| Standard LM | 51,575  | 144,795 | 270,789 |
| DLM       | 40,704  | 2,135,429 | 8,502,075 |

Table 5 WER before/after applying DLM in CSJ.

|          | 1-best | DLM   |
|----------|--------|-------|
| A-Eval   | 18.0   | 16.9  |
| O-Eval   | 36.3   | 34.2  |

Figure 1 Relationship of shrunken model size and WER between the model shrinkage methods in Eval-Small.

Figure 2 Shrunken model size and WER with our proposed methods in Eval-Small and Eval-Large.

for shrinking the DLM, i.e. for MSP training, set A-Small is used under condition Eval-Small. Under condition Eval-Large, a large data set, A-Large, is used for this purpose. Condition Eval-Cross is prepared for evaluating the model shrinkage in a cross-domain condition. The evaluation set consists of omnibus lectures, while DLM training and MSP training are performed with academic lectures. In condition Eval-Out, omnibus lectures are used for both MSP training and evaluation. This is to confirm whether or not MSP training performs properly even in an out-of-domain task.

Figure 1 shows the relationships between the size of the shrinkage models and the WER under condition Eval-Small. ‘Power’, ‘MI’, ‘SBS’, ‘Proposal (w/ scaling)’ and ‘Proposal (w/o scaling)’ denote the power based cut-off, MI approach, SBS method, our proposed method and its special case method where $b_k = 1$ for all $k$, respectively. Results of the shrinkage models over 500 kilo parameters are omitted from the figure, because the WER differences were very small. This means the original DLM contains many redundant parameters, and they can be removed properly even by using the power based cut-off.

The power based cut-off and MI approach provided higher error rates than SBS and our proposed methods. The power based cut-off does not have a framework to keep error rate low. The MI approach decides the permutation of parameters without taking the original model parameters into account. However, small models shrunk by the MI approach performed at the same level as our proposed methods. We believe that, since MI can be regarded as a discriminative criterion, the MI approach could obtain some of the most important parameters.

SBS and our proposed methods performed at the same level as the original model up to a size of about 100 kilo size. As mentioned above, SBS minimizes Eq. (52) directly and our proposed methods minimize its approximation, i.e. Eq. (12). This result shows the validity of this approximation.

A comparison of our proposed method and its special case, i.e. Proposal (w/ scaling) and Proposal (w/o scaling), revealed that the special case method performed slightly better than the original method. In our experiments, we evaluated the model shrinkage methods in DLM task for speech recognition. In this case, the standard LM provides powerful help to the DLM as regards word prediction. This potentially affects the model shrinkage performance. If our methods are employed for different tasks such as tagging and parsing, the original method might outperform the special case method, since the former includes the latter. In future work, we need to investigate this point in different tasks.

Figure 2 shows results of our proposed methods under different conditions in terms of the amount of data used for MSP training. The amount of data did not affect the model shrinkage to any the significant degree. Our proposed methods assign parameters that correspond to no appearance fea-
tures in training data to the end of the permutation. However, the small set A-Small, about 1000 utterances, was effective in terms of choosing important parameters.

Figure 3 shows results for set O-Eval. Our proposed method, Proposal (w/o scaling), succeeded in shrinking the model with little WER degradation even in an out-of-domain task. The success in condition Eval-Cross shows that we can use the model shrinkage function in a variety of domains even if MSP training is undertaken with a specific domain data set. And the result in condition Eval-Out shows that the model shrinkage information can be obtained properly only if we prepare a small data set, even if its domain does not match that of the DLM training data.

In addition, we confirmed that Proposal (w/o scaling) and SBS also performed properly at the same level. From Figs. 2 and 3, we can see that important features commonly appear in several data sets.

To confirm the consistency of the results obtained under CSJ, we also evaluated the model shrinkage methods with the MIT Lecture Corpus (MITLC), which is an English lecture corpus. We divided the lectures into the three sets shown Table 6 and the other lectures. The latter is used for training a standard LM for the SOLON. We employed a word 4-gram model. The lexicon size was 70,398. The AM was trained using the training data for the standard LM and set M-Large in Table 6.

We generated 5000-best lists of all the utterances shown in Table 6. Sets M-Large and M-Small are training and development sets for DLM, respectively. Set M-Eval is an evaluation set. The features are counts of the word unigrams, bigrams and trigrams and POS unigrams, bigrams and trigrams. Table 7 shows the number of parameters between the standard LM and DLM. The WERs before and after applying the DLM were 37.0% and 35.2%, respectively (see Table 8).

MSP training was undertaken using set M-Small. Thus, this experiment corresponds to condition Eval-Small in Table 3. The results are shown in Fig. 4.

When comparing Figs. 1 and 4, the same result was observed as regards the power based cut-off, SBS and our proposed methods. In contrast, the MI approach performed better than in the CSJ. We believe that, since the MI approach decides the permutation of parameters without seeing the parameter values of the DLM, its result depends on the corpora.

5. Conclusion

We described the model shrinkage problem (MSP) in this paper. The purpose is to enable application developers and users to control model size and thus improve the usability of ASR systems. We provided a practicable formulation for the MSP and proposed two types of model shrinkage methods for linear models. Our proposed methods decide the permutation of important parameters taking the simple statistics and model parameter values into account. We also converted the MI approach and SBS, which were developed for the feature selection problem, into the MSP and compared them with our proposed methods. Our experiments were designed to simulate certain practical scenarios. They included cases where the data for DLM training consisted of the data for the model shrinkage, and where different data sets were given. Then, we prepared different evaluation sets and corpora. In all cases, our proposed methods were able to shrink models considerably with little degradation in accuracy. Our proposed methods also outperformed the MI approach and were comparable with SBS, which took many more hours.
to decide the permutation of parameters than our proposed methods.

We described experiments only for the DLM of ASR, but model shrinkage can be employed for other models. However, the results might differ when shrinking the DLM with other models. This should be investigated in future work.

Acknowledgment

We thank the MIT Spoken Language Systems Group for helping us to perform speech recognition experiments based on the MITLCC.

References

[1] Y.Y. Wang, D. Yu, Y.C. Ju, and A. Acero, “An introduction to voice search,” IEEE Signal Process. Mag., vol.25, pp.28–38, 2008.
[2] J. Choi, D. Hindle, J. Hirschberg, I. Magrin-chagnolleau, C. Nakatani, F. Pereira, O. Pereira, A. Singhal, and S. Whittaker, “An overview of the AT&T spoken document retrieval,” Proc. Euraisp Journal on Applied Signal Processing 2003, no.2, 1 Feb. 2003 12 1998 DARPA Broadcast News Transcription and Understanding Workshop, pp.182–188, 1998.
[3] J. Mamou, D. Carmel, and R. Hoory, “Spoken document retrieval from call-center conversations,” Proc. 29th annual international ACM SIGIR conference on research and development in information retrieval, pp.51–58, 2006.
[4] Z.Y. Zhou, P. Yu, C. Chelba, and F. Seide, “Towards spoken-document retrieval for the Internet: Lattice indexing for large-scale web-search architectures,” Proc. Human Language Technology Conference (HLT/NAACL), pp.51–58, 2006.
[5] R. Higashinaka, N. Miyazaki, M. Nakano, and K. Aikawa, “Evaluating discourse understanding in spoken dialogue systems,” ACM Trans. Speech Language Process., vol.1, no.1, pp.1–20, 2004.
[6] M.F. McTear, “Spoken dialogue technology: Enabling the conversational user interface,” ACM Computing Surv., vol.34, no.1, pp.90–169, 2002.
[7] A.I. Rudnicki, E.I. Rudnick, C. Bennett, A.W. Black, A. Chotomongcol, K. Lenzo, A. Oh, and R. Singh, “Task and domain specific modelling in the Carnegie Mellon communication system,” Proc. ICSLP, pp.130–134, 2000.
[8] K. Sudoh, K. Duh, and H. Tsukada, “NTT statistical machine translation system in IWSLT 2010,” Proc. International Workshop on Spoken Language Translation, pp.147–152, 2010.
[9] T. Takekawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, “Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world,” Proc. LREC, pp.147–152, 2002.
[10] B. Roark, M. Saracacl, M. Collins, and M. Johnson, “Discriminative language modeling with conditional random fields and the perceptron algorithm,” Proc. ACL, pp.47–54, 2004.
[11] B. Roark, M. Saracacl, and M. Collins, “Discriminative n-gram language modeling,” Comput. Speech Language, vol.21, no.2, pp.373–392, 2007.
[12] Z. Zhou, J. Gao, F.K. Soong, and H. Meng, “A comparative study of discriminative methods for reranking LVCSR n-best hypotheses in domain adaptation and generalization,” Proc. ICASSP, pp.141–144, 2006.
[13] M. Collins, B. Roark, and M. Saracacl, “Discriminative syntactic language modeling for speech recognition,” Proc. ACL, pp.507–514, 2005.
[14] A. Kobayashi, T. Oku, S. Homma, S. Sato, T. Imai, and T. Takagi, “Discriminative rescoring based on minimization of word errors for transcribing broadcast news,” Proc. Interspeech, pp.1574–1577, 2008.
[15] T. Oba, T. Hori, and A. Nakamura, “A comparative study on methods of weighted language model training for reranking LVCSR n-best hypotheses,” Proc. ICASSP, pp.5126–5129, 2010.
[16] V. Magdin and H. Jiang, “Large margin estimation of n-gram language models for speech recognition via linear programming,” Proc. ICASSP, pp.5398–5401, 2010.
[17] I. Guyon, “An introduction to variable and feature selection,” J. Machine Learning Research, vol.3, pp.1157–1182, 2003.
[18] F. Jelinek, “Self-organized language modeling for speech recognition,” in Readings in Speech Recognition, pp.450–506, Morgan Kaufmann, 1990.
[19] A. Stolcke, “Entropy-based pruning of backoff language models,” Proc. DARPA News Transcription and Understanding Workshop, pp.270–274, 1998.
[20] J. Li, H. Wang, D. Ren, and G. Li, “Discriminative pruning of language models for Chinese word segmentation,” Proc. Association for Computational Linguistics, pp.1001–1008, 2006.
[21] T. Oba, T. Hori, A. Nakamura, and A. Ito, “Round-robin duel discriminative language models,” IEEE Trans. Audio Speech Language Process., vol.20, no.4, pp.1244–1255, 2012.
[22] C.D. Manning, P. Raghavan, and H. Schuetze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[23] F. Fleuret, “Fast binary feature selection with conditional mutual information,” J. Machine Learning Research, vol.5, pp.1531–1555, 2004.
[24] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.8, pp.1226–1238, 2005.
[25] A.W. Whitney, “A direct method of nonparametric measurement selection,” IEEE Trans. Comput., vol.20, no.9, pp.1100–1103, 1971.
[26] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, “Spontaneous speech corpus of Japanese,” Proc. ICLRE, pp.947–952, 2000.
[27] T. Hori and A. Nakamura, “Generalized fast on-the-fly composition algorithm for WFST-based speech recognition,” Proc. Interspeech, pp.284–289, 2005.
[28] R. Kneser and H. Ney, “Improved backoff for m-gram language modeling,” Proc. ICASSP, pp.181–184, 1995.
[29] B. Roark, M. Saracacl, and M. Collins, “Corrective language modeling for large vocabulary ASR with the perceptron algorithm,” Proc. ICASSP, pp.749–752, 2004.
[30] E. McDermott, T.J. Hazen, J.L. Roux, A. Nakamura, and S. Katagiri, “Discriminative training for large vocabulary speech recognition using minimum classification error,” IEEE Trans. Audio Speech Language Process., vol.15, no.1, pp.203–223, 2007.

Takanobu Oba received B.E. and M.E. degrees from Tohoku University, Sendai, Japan, in 2002 and 2004, respectively. Since 2004, he has been engaged in research on spoken language processing at the NTT Communication Science Laboratories, Kyoto, Japan. He received the 25th Awa Kiyoshi Science Promotion Award from the Acoustical Society of Japan (ASJ) in 2008. He received Ph.D. (Eng.) degree from Tohoku University in 2011. He is an affiliate mem-

ruber of the Institute of Electrical and Electronics Engineers (IEEE) and the ASJ.
Takaaki Hori received B.E. and M.E. degrees in electrical and information engineering from Yamagata University, Yonezawa, Japan, in 1994 and 1996, respectively, and was awarded a Ph.D. in system and information engineering by Yamagata University in 1999. Since 1999, he has been engaged in research on spoken language processing at the NTT Cyber Space Laboratories, Yokosuka, Japan. He was a Visiting Scientist at the Massachusetts Institute of Technology, Cambridge, from 2006 to 2007. He received the 22nd Awaya Kiyoshi Science Promotion Award from the Acoustical Society of Japan (ASJ) in 2005. He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the ASJ.

Atsushi Nakamura received B.E., M.E., and Dr.Eng. degrees from Kyushu University, Fukuoka, Japan, in 1985, 1987 and 2001, respectively. In 1987, he joined Nippon Telegraph and Telephone Corporation (NTT), where he engaged in the research and development of network service platforms, including studies on the application of speech processing technologies to network services, at Musashino Electrical Communication Laboratories, Tokyo, Japan. From 1994 to 2000, he was with Advanced Telecommunications Research (ATR) Institute, Kyoto, Japan, as a Senior Researcher, working on the research of spontaneous speech recognition, the construction of spoken language database and the development of speech translation systems. Since April, 2000, he has been with NTT Communication Science Laboratories, Kyoto, Japan. His research interests include acoustic modeling of speech, speech recognition and synthesis, spoken language processing systems, speech production and perception, computational phonetics and phonology, and application of learning theories to signal analysis and modeling. Dr. Nakamura is a senior member of the IEEE and a member of the Acoustical Society of Japan (ASJ). He also serves as a Vice Chair of the IEEE Signal Processing Society Kansai Chapter. He received the IEICE Paper Award, and the Telecom-technology Award of The Telecommunications Advancement Foundation, in 2004 and 2006, respectively.

Akinori Ito was born in Yamagata, Japan, in 1963. He received B.E., M.E., and Ph.D. degrees from Tohoku University, Sendai, Japan, in 1984, 1986, and 1992, respectively. Since 1992, he has worked with Research Center for Information Sciences and Education Center for Information Processing, Tohoku University. He was with the Faculty of Engineering, Yamagata University, from 1995 to 2002. From 1998 to 1999, he worked with the College of Engineering, Boston University, MA, USA, as a Visiting Scholar. He is now a Professor of the Graduate School of Engineering, Tohoku University. He is engaged in spoken language processing, statistical text processing, and audio signal processing. He is a member of the Acoustical Society of Japan, the Information Processing Society of Japan, and the IEEE.