Outlier Detection for Acoustic Model Training Using Robust Statistics

Shigeki Matsuda, Wolfgang Herbordt, Satoshi Nakamura

Department of Acoustics and Speech Research, Spoken Language Translation Research Labs, Advanced Telecommunications Research Institute International, Kyoto, Japan

{shigeki.matsuda, wolfgang.herbordt, satoshi.nakamura}@atr.jp

Abstract

In this paper, we propose an acoustic model training technique which is robust against outliers such as clipping, unexpected noise, poorly pronounced word segments, or mistranscriptions, which deteriorate the quality of the acoustic models and in turn decrease speech recognition performance. The outlier-robust acoustic model training technique is based on a maximum likelihood (ML) criterion and automatically detects and removes outliers from the training data. Experiments with artificially contaminated mis-transcribed training data show that nearly the same word error rate can be obtained for contaminated data using the proposed technique as for uncontaminated data. Application to a dialogue speech database with unknown outliers reduces the errors by 4.03%.

1. Introduction

In recent years, the amount of data for training acoustic models has increased greatly to improve robustness against variables like speaking style, dialects, or accents. Therefore, large databases of speech such as spontaneous speech or lecture speech have been collected. However, to train accurate acoustic models, not only is speech recording necessary, but the database should also be cleaned of unexpected noise, poorly pronounced words, or incorrect transcriptions. Since the manual cleaning of large databases is expensive and time-consuming, it is desirable to reduce the cost by using less accurate databases and to clean the training data automatically during the acoustic model training.

Model training techniques using speech databases with mistranscriptions were proposed in [1, 2, 3, 4]. The lightly supervised training technique [4] was applied to speech data of broadcast news with CC (Closed Caption) transcripts, which are known to contain many mis-transcriptions. This technique selects those phrases of three or more contiguous words for which the CC transcripts and the hypotheses of the decoder agree.

In this work, we present a method for not only detecting and removing wrongly transcribed data from the acoustic model training, but all kinds of outliers, such as clipping, unexpected noise, or poorly pronounced word segments. The proposed technique operates on an HMM (Hidden Markov Model) state level or on a phoneme level, thus potentially minimizing the amount of removed training data. The method is based on the MCD (‘Minimum Covariance Determinant’) [5, 6], an outlier-robust estimator of mean and covariance of multivariate random variables. Outlier-robustness signifies in this context insensitivity to small deviations from the assumptions [7]. The proposed technique is evaluated using both artificially contaminated training data and real training data.

This paper is organized as follows: In Sect. 2, we review the MCD algorithm, and propose the ML-based outlier-robust estimation technique. In Sect. 3, experimental results are presented. Section 4 concludes the paper.

2. Maximum Likelihood-Based Outlier Detection Technique

2.1. Minimum Covariance Determinant

MCD was proposed in [5, 6] as a technique for robust estimation of mean \(\mu_u\) and covariance \(\Sigma_u\) of a multivariate random variable. Assume \(n\) observations \(X = \{x_1, \cdots, x_n\}\) of a multivariate random variable contain a known number \(n - h\) of outliers. Then, the MCD algorithm iteratively finds those observations \(H_u \subset \{1, \cdots, n\}\), where \(|H_u| = h\), whose covariance matrix has the lowest determinant (Fig. 1). The remaining \(n - h\) observations are considered as outliers and are not used in estimating the mean vector or the covariance matrix, thus yielding an outlier-robust estimate. In addition, \(u\) denotes the number of iterations, and \((\cdot)'\) is the transposition of a vector or matrix.

This MCD algorithm can be applied easily to HMMs that are widely used as acoustic models for speech recognition. If each HMM state has a single Gaussian distribution, the mean \(\mu_u\) and the variance \(\Sigma_u\) completely describe the Gaussian distribution, and the observation set \(X\) corresponds to the feature vectors which are allocated to an HMM state by using the Viterbi algorithm or the forward algorithm.
Step 1: Estimate distribution parameters

\[
\mu_u := \frac{1}{n} \sum_{k \in H_u} x_k
\]

\[
\Sigma_u := \frac{1}{n} \sum_{k \in H_u} (x_k - \mu_u) (x_k - \mu_u)'
\]

Step 2: Calculate Mahalanobis distances

If \( \text{det}(\Sigma_u) \neq 0 \), for \( k = 1, \ldots, n \)

\[
d_u(k) := \sqrt{(x_k - \mu_u)' \Sigma_u^{-1} (x_k - \mu_u)} \tag{1}
\]

Step 3: Sort training samples

\[
\{d_u(k); k \in H_{u+1}\} := \{(d_u)_1, \ldots, (d_u)_n\}
\]

where \((d_u)_1 \leq \cdots \leq (d_u)_h \cdots \leq (d_u)_n\)

Go to Step 1

2.2. Maximum Likelihood-Based Outlier-Robust Estimation

In HMMs, the output probability density of each HMM state is represented by several (diagonal) distributions in order to represent speech signals efficiently. Obviously, the MCD algorithm may be directly applied for the parameter estimation of each mixture component by categorizing the training samples (feature vectors). This straightforward approach is suboptimum for two reasons: (1) Outliers for one training sample should be detected with respect to all mixture components simultaneously and not separately. (2) The correlation between sequences of feature vectors should be exploited.

For these reasons, we propose to use the HMM output probability of feature vector sequences for detecting outliers in, e.g., HMM states or HMM state sequences: Let \( \Theta_i \) defines the output distribution and the transition probabilities of the \( i \)-th HMM state. Training data for estimating the output distribution of the \( i \)-th HMM state is obtained by segmentation with the Viterbi algorithm. The training data consists of multiple feature vector sequences, the \( k \)-th feature vector sequence in the training data can be described by \((x_{k,1}, x_{k,2}, \ldots, x_{k,T_k})\). A measure for the distance between the model \( P \) to the \( k \)-th feature vector sequence can then be calculated as follows:

\[
d(k) := P((x_{k,1}, x_{k,2}, \ldots, x_{k,T_k})|\Theta_i) \]

The set of the feature vector sequences, that includes \( h \) feature vectors, with the highest product of probabilities \( d(k) \), are chosen as training samples. The selection of \( h \) out of \( n \) feature vectors with the constraint of choosing only from among feature vector sequences of given length \( T_k \) is identical to the knapsack problem in computer science, and can be solved by dynamic programming.

The proposed technique can be flexibly applied to training with multiple databases. Some databases are known not to contain any outliers. It is not necessary to detect outliers from these clean databases. Outlier detection performance may be improved by removing outliers from only additional databases that contain outliers. In case that the proposed technique is applied to the joint set of all databases, I) not only training samples of the contaminated additional databases but also of the clean databases may be removed (Fig. 2-I). On the other hand, in the case II) that the proposed technique is applied only to the contaminated additional training data, the clean training data is used unconditionally (Fig. 2-II).

3. Outlier Detection Experiments

In this section, we experimentally investigate the efficiency of the outlier detection.

3.1. Experimental Conditions

The ATRASR version 3.1 developed by ATR – Spoken Language Translation Laboratories is used as a large vocabulary speech recognition system. This ASR system is a two-pass decoder. A feature vector consists of 12 MFCCs, 12 ΔMFCCs and Δpow extracted from frames of 20-ms length with 10-ms frame shift. CMS (Cepstrum Mean Subtraction) is applied to the feature vectors. Acoustic models were trained using five hours dialogue speech from the ATR travel arrangement task database and 25 hours real speech of phonetically balanced sentences. Each HMM state has five Gaussian mixture components with diagonal covariance matrices. For testing, we used the basic travel expression corpus (BTEC) [11] testset-01 (510 sentences, four males and six females, each speaker uttered 51 sentences). The system uses a word bi-gram language model and a composite word tri-gram language model [9]. Each language model is trained from the spontaneous speech database (SDB), language database (LDB) and spoken language database (SLDB), with a total number of 6.1M words [10]. The lexicon size is 34k words.

In our experiments, we evaluate A) state-level outlier detection on feature vectors, B) state-level outlier detection on feature vector sequences, and C) context-dependent phoneme level outlier detection on feature vector sequences. In Method A, the feature vectors of the training data are pooled to HMM states using Viterbi alignment, and the outlier detection is applied to each HMM state by applying the ML-based outlier detection to individual feature vectors \( (T_k = 1, \forall k) \). Method B detects outliers in the pooled training data in each HMM state using the feature vector sequences for the decision as they were found by the Viterbi alignment. For Method C, feature vector sequences are pooled into context-dependent phonemes, and the outlier
In the first experiment, a state-tying structure with 1,400 states is generated using ML-SSS (successive state splitting) [8] with half of the training data. The other half of the training data contains 10% artificial mis-transcriptions, yielding 5% artificial mis-transcriptions in the whole training data. A Japanese syllable consists of vowels (V), consonant-vowels (CV) and CCV. Mis-transcriptions are generated by changing from a consonant to another consonant or to a vowel and by changing from a vowel to another vowel. Non-existent phoneme sequences, such as CCC, are avoided. These sets of training data are used as a clean database and an additional database respectively.

Fig. 3 illustrates the performance, measured as ‘outlier detection rate’, as a function of the ‘reduction rate’ $\epsilon = (n - h)/n \cdot 100\%$ of the outlier detection algorithms. The $\epsilon = 0\%$ reduction rate means that all the training data is used for training and corresponds to conventional acoustic model training. The same reduction rate is applied to all statistical models so that a uniform distribution of the outliers over the training data is assumed. It can be seen that the number of detected outliers increases with increasing reduction rate $\epsilon$. With II) outlier detection on the additional training data, higher detection performance is obtained than by using I) the whole training data. For II), the outlier detection rate is about 40% for $\epsilon = 5\%$ while only 20 – 30% for I). The highest outlier detection rate is obtained with Method B-II. Note that – due to erroneous outlier detection – an increasing outlier reduction rate also means an increase of falsely removed clean training data.

Moreover, to evaluate robustness of acoustic modeling to mis-transcribed training data, we calculate the distance between an acoustic model trained using only correctly transcribed data and an acoustic model estimated by using each proposed outlier detection technique. This distance is obtained by calculating the average Bhattacharyya distance between HMM states of all acoustic models. This distance is illustrated in Fig. 4. Method B-II generates acoustic models that are nearest to the acoustic model trained using only correctly transcribed training data, thus, increases from Method A to Method C.

3.2. Experimental Results

3.2.1. Artificially Contaminated Training Data

In the first experiment, a state-tying structure with 1,400 states is generated using ML-SSS (successive state splitting) [8] with half of the training data. The other half of the training data contains 10% artificial mis-transcriptions, yielding 5% artificial mis-transcriptions in the whole training data. A Japanese syllable consists of vowels (V), consonant-vowels (CV) and CCV. Mis-transcriptions are generated by changing from a consonant to another consonant or to a vowel and by changing from a vowel to another vowel. Non-existent phoneme sequences, such as CCC, are avoided. These sets of training data are used as a clean database and an additional database respectively.

The distance is minimal at around a 15% reduction rate. Since a large amount of clean data may be removed with a higher reduction rate, the distance may increase.

In Fig. 5 and Fig. 6, the word error rate for outlier-robust acoustic model estimation is compared to conventional ML estimation ($\epsilon = 0\%$) as a function of $\epsilon = 0 \ldots 5\%$. For ML estimation using the correctly transcribed and the mis-transcribed training data, the word error rate is 8.62%; using only the correctly transcribed data, the word error rate is 8.20%. Thus, by using mis-transcribed data, the word error rate is increased by 0.42%. We can observe that outlier detection on the II) additional training data yields lower word error rate than outlier detection on the I) whole training data. These results conform to the detection rates (Fig. 5) and to the average Bhattacharyya distance (Fig. 4). For all methods, the similar general behavior can be observed: The word error rate decreases with increasing $\epsilon$ until a minimum around $\epsilon = 3\%$ is reached, and increases for $\epsilon > 3\%$. This behavior results from the fact that with increasing $\epsilon$, not only does the number of detected outliers increase but also the total amount of training data decreases. For the same reason, the minimum is not reached for $\epsilon = 5\%$, but for the smaller reduction rate of $\epsilon = 3\%$. For Method B, nearly the same word error rate is obtained as for excluding the mis-transcribed data from the training data.

3.2.2. ‘Naturally’ Contaminated Training Data

In the second experiment, we assume that the training data itself contains outliers such as poorly pronounced phonemes or mis-transcriptions. Therefore, outlier detections on the I) whole training data are evaluated only. The whole training data are used in an initial phase to generate the state-tying structure with 1,400 states using ML-SSS and during the parameter training, where outlier detection is applied. The results are depicted in Fig. 7 for $\epsilon = 0 \ldots 2\%$. The reduction rates $\epsilon$ are lower than those in the first experiment since the training data only contains ‘natural’ outliers, which are assumed to be less frequent. It can be seen that, by using outlier-robust acoustic model estimation, the word error rate improves from 7.22% ($\epsilon = 0\%$) to 6.94% ($\epsilon = 1.0\%$, Method B). As in the first experiment, Method B provides the best performance.
4. Conclusion

We presented a framework for acoustic model training to automatically detect outliers such as mis-transcriptions, poorly pronounced phonemes, or unexpected noise. We implemented the outlier-robust training procedure for detecting outliers based on feature vectors and feature vector sequences at an HMM state level and on a context-dependent phoneme level. Experimental results show that outliers can be efficiently removed using feature vector sequences on an HMM state level: For artificially contaminated data, almost the same word error rate is obtained as for directly training on the clean data. For ‘naturally’ contaminated data, the word error rate is noticeably decreased relative to conventional non-robust acoustic model training. In future work, we will analyze outlier-robust acoustic model training more fundamentally and for other types of outliers. Further, we will apply the training procedure to databases which are known to contain large numbers of outliers.

5. Acknowledgements

The research reported here was supported in part by a contract with the National Institute of Information and Communications Technology entitled ‘A study of speech dialogue translation technology based on a large corpus’.

6. References