The following full text is a publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/166330

Please be advised that this information was generated on 2018-11-23 and may be subject to change.
Analytical assessment of dual-stream merging for noise-robust ASR

Louis ten Bosch\textsuperscript{1}, Bert Cranen\textsuperscript{1}, Yang Sun\textsuperscript{2}

\textsuperscript{1}Radboud University Nijmegen, The Netherlands
\textsuperscript{2}Nuance Communications, Aachen, Germany

\texttt{l.tenbosch@let.ru.nl, b.cranen@let.ru.nl, jonathan.eric.sun@gmail.com}

Abstract

In previous studies (on Aurora), it was found that merging a posteriori probability streams from different classifiers (GMM, MLP, Sparse Coding) can improve the noise robustness of ASR. Maximizing word accuracy required the stream weights to be systematically dependent on the specific input streams and SNR. The tuning of the weights, however, was largely a matter of trial and error and typically involved a laborious grid search. In this paper, we propose two fundamental, analytical methods to better understand these empirical findings. To that end, we maximize the trustworthiness of merged streams as function of the stream weights. Trustworthiness is defined as the probability that the winning state in a probability vector correctly predicts a golden reference state obtained by a forced alignment. Even though our approach is not directly equivalent to optimizing word accuracy, both methods appear highly useful to obtain insight in stream properties that determine the success of a given merge (or the lack thereof). Furthermore, both methods clearly support the trends that exist in the grid-search based empirical observations.

**Index Terms:** stream merging, noise robustness, speech recognition, analytical methods, trustworthiness

1. Introduction

Designing novel ASR systems that can outperform state-of-the-art systems in both clean and noisy conditions is challenging. Systems that show excellent performance in certain conditions often underperform in other conditions. For example, the results in [1] show that a Sparse Classification (SC) system operating on Mel-band spectra can outperform a traditional MFCC-based Gaussian Mixture Model-based (GMM) system in very noisy conditions; in clean conditions, the situation is reversed though.

Different front ends may exhibit different strengths and weaknesses. It is therefore attractive to try to improve recognition performance by exploiting their complementary properties. In the past, numerous approaches to combine information from different feature streams and/or classification strategies have been proposed. Typically, one discards fusion at the level of features [2, 3, 4, 5], at the level of probabilities [6, 7, 8, 9, 10], and at the level of hypotheses [11, 12, 13, 14].

This paper focuses on a novel account of stream merging at the probability level. We define a stream as a classifier output that consists of sequences of probability vectors (‘frames’), each frame representing a probability distribution on a set of Hidden Markov Model (HMM) states. When merging the frames of different classifiers, we limit ourselves to the weighted product approach: The elements of the respective stream probability vectors are raised to an exponent (the stream weight), after which the weighted input vectors are multiplied element-wise to yield a new, single merged stream output vector. The resulting vector is then fed to a classical Viterbi decoder back end.

Generally, equal weights are not the optimal choice to get optimal recognition performance. In situations where similar Multi-layer Perceptron (MLP) based classifiers are combined (e.g., in [15] and [16]) MLP classifiers trained on different acoustic features were combined), it has been shown that a frame-wise, between-stream comparison of the inverse or minimum entropy of the probability vectors provides a useful approximation for the stream weight [17]. However, when two completely different types of classifiers must be combined, intrinsic quality differences between the classifiers that generate the streams, may cause substantial differences in the entropy. In such situations, inverse entropy is less likely to constitute an adequate descriptor of the “trustworthiness” of a stream, and different stream weighting schemes might be required.

In this paper we present a novel analytical account of how to merge two streams that stem from two entirely different types of classifiers, namely an MLP classifier [18] and a Sparse Coding classifier [1]. Previous research on the AURORA-2 task [19] has shown that, in contrast to an MLP classifier which is trained to assign the bulk of the probability mass to a unique state, the SC system tends to divide the probability mass relatively evenly over a number of states that are acoustically similar [20, 21]. Using a plain grid search approach, it was found that optimal stream combination required different, SNR dependent stream weights in order to achieve the highest word accuracies after decoding [22]. In practice we found that the relative contribution of the SC stream needed to be systematically and gradually increased for decreasing SNRs.

Despite the fact that these observations seem quite logical given the SNRs at which the individual SC and MLP classifiers perform best, it is difficult to get theoretical insight about which stream properties predominantly determine the success of a given merging scheme. The aim of the current paper is to present analytical tools that facilitate understanding the underlying mechanisms which allow the Viterbi decoder to do a better job in finding the correct \{frame, state\}-path. We will illustrate our ideas by analyzing the same SC and MLP streams derived from the AURORA-2 data as were used in previous experiments (cf. [18]).

2. Experimental set up

2.1. Speech data

For the experiments in this paper we used the AURORA-2 speech database. AURORA-2 contains sequences of up to seven connected digits from the 11-digit set \{oh, zero, one, \ldots, nine\} corrupted by eight different types of additive noise at seven dif-
ferent noise levels (i.e. clean and SNR = 20, 15, 10, 5, 0, -5 dB) [19]. As in most studies on AURORA-2, we model each digit as a sequence of 16 consecutive HMM states, while silence is represented by a model using three consecutive states. In total, all models comprise \((11 \times 16 + 3 = 179)\) states.

2.2. SC and MLP input streams

In the front-end of our recognizer, we apply two different classifiers (MLP, SC). Each classifier produces a 179-dimensional posterior state probability vector, which is updated every 10 ms. Subsequently, the outputs of the classifiers are combined to yield a new stream of posterior probability estimates (also 179-dimensional vectors), which are processed by a Viterbi decoder back-end (implemented in MATLAB).

The MLP classifier is a discriminative classifier, which has been widely used for acoustic modeling as an alternative for the Gaussian mixture model (GMM) [23]. Due to the discriminative nature of the training, the output vectors of an MLP classifier tend to attribute most of the probability mass to a single state. The MLP system used here was trained using the Quicknet software [24]. The input consists of 13 perceptual linear prediction cepstral coefficients and their corresponding first and second order time derivatives (39 coefficients in total) combined to span a temporal context of 90 ms (9 frames). For building the MLP system, the multi-condition training set in AURORA-2 was split into a set of 7685 utterances for optimizing the MLP parameters and 755 utterances for cross-validation. The MLP had one hidden layer, the optimal size of which was determined based on the frame accuracy obtained on the cross-validation set.

The SC classifier, by contrast, approximates energy spectrogram representations of speech segments as a sparse, non-negative, linear combination of exemplar spectrograms with a duration of 300 ms taken from two dictionaries (one consisting of speech exemplars taken from the set of 7685 utterances for training the MLP system, the other of exemplars of the noise in the multi-condition training set). All frames in the speech exemplar dictionary are labeled as one of the 179 states that make up all models. Using the weighting coefficients of the speech exemplars found in the linear decomposition (i.e., the speech activation scores), each frame in a segment of speech input can be associated with a vector of posterior state probabilities. In practice, the probability mass usually appears to get distributed over more than one element of the output vectors of the SC classifier. The applied SC system is described in detail in [1].

2.3. Weighted stream merging

In our stream merging approach, we take the traditional single stream Viterbi decoding as a starting point. The optimal word sequence \(W\) is the one that maximizes the total likelihood across all possible word sequences \(O\) given the observations \(O\):

\[
W = \arg \max_W P(W|O) = \arg \max_{t=1}^T \left( \prod_{t=1}^{T} p(o_t|s_t) P(s_t|s_{t-1}) \right) \cdot P(W)
\]

where \(s_t\) denotes the state occupied at time \(t\), \(o_t\) the observed feature vector at time \(t\) and \(s_t \subset W\) is a short hand notation for all admissible paths that represent a valid word sequence. Note that the rewrite of eq. (2) into eq. (3) is to account for the fact that our classifiers produce posterior probability estimates.

Using eq. (3), as a reference, we implement stream merging by replacing \(\hat{P}(s_t|o_t)\) by a weighed product of posterior probability estimates that are associated with our two input streams:

\[
\hat{P}(s_t|o_t) = P_{\text{mlp}}(s_t|o_t) + \alpha P_{\text{sc}}(s_t|o_t)\quad(4)
\]

Substitution of Eq. 4 in Eq. 3 does not directly provide a handle to analytically compute the stream weights that minimize the WER, neither globally (frame-independent) nor locally (i.e. frame-dependent). Therefore, rather than assessing the competitive power of streams directly at the WER level, we have opted for an alternative strategy: Assuming that the observed word accuracy after the Viterbi decoding is strongly related to the average frame-based trustworthiness of the input stream, we attempt to optimize the 'trustworthiness' of a merged stream at the frame level. The trustworthiness of a stream is defined as the probability of a winning state being correct, in the sense of equal to the winning state in a golden reference stream. This reference stream consists of probability distribution vectors of which only one state has probability 1. It is constructed by a conventional forced alignment procedure, in which the underlying clean utterances are aligned with the correct acoustic model sequence.

In the next section, we will elaborate on two analytical methods that attempt to optimize the trustworthiness of a merged stream \(P_{\text{mlp}}^{\alpha} \cdot P_{\text{sc}}^{(1-\alpha)}\) as a function of the weight \(\alpha\).

3. Trustworthiness optimization

We will describe two methods. The first method is based on the Kullback-Leibler dissimilarity between the golden reference stream (denoted \(P_G\)) and the merged stream \(P_{\text{mlp}}^{\alpha} \cdot P_{\text{sc}}^{(1-\alpha)}\). The second method is based on a regularized algebraic decomposition method that finds an optimal linear matrix combination of \(P_{\text{mlp}}\) and \(P_{\text{sc}}\) to reconstruct \(P_G\).

3.1. Kullback-Leibler dissimilarity

In the first method, we compare \(P_{\text{mlp}}^{\alpha} \cdot P_{\text{sc}}^{(1-\alpha)}\) with the golden reference \(P_G\), by using the average frame-based symmetrized KL dissimilarity as distance measure. The symmetrized KL dissimilarity between two probability vectors \(p\) and \(q\) with elements \(\{p_j\}\) and \(\{q_j\}\) (\(j\) indicating the state id) reads:

\[
\text{KL}(p, q) = 1/2 \sum_j (p_j - q_j) \log(p_j/q_j)
\]

Figure 1 depicts the KL dissimilarity, averaged over all frames, between merged and golden reference stream (left two panels) and WER (on a log scale) after Viterbi decoding (rightmost panel) as a function of \(\alpha\). A value of \(\alpha = 0\) corresponds with a merged stream only containing SC info, and \(\alpha = 1\) only containing MLP info. Each color corresponds to a particular SNR in one of the data sets of AURORA-2. The curves in the left panel correspond to training data; the solid lines in the mid panel to test set A and the dashed curves to test set B. In the right panel, the observed WERs after Viterbi decoding is displayed.

We observe that the KL dissimilarity is larger for the test sets than for the training set. It also increases when SNR decreases; moreover, for decreasing SNR the difference between
With our second method, we search two 179-by-179 matrices. The dependencies are clearly visible.

ever, is unable to precisely predict the stream weights; only ten times, which was sufficient to reach convergence.

We first randomly initialized \( \hat{B} \) and subsequently iterated eqs. 10 and 11 ten times, which was enough to achieve a lag-2 correctness of about 90 percent. The Lyapunov equations 10 and 11 can be solved iteratively.

\[ A \cdot P^t + B \cdot Q^t - QP^t = \lambda_1 (A + B - 1) ee^t \]  
\[ A \cdot PQ^t + B \cdot QQ^t - QP^t = \lambda_2 (A + B - 1) ee^t \]  

The Lyapunov equations 10 and 11 can be solved iteratively. We first randomly initialized \( \hat{B} \) and subsequently iterated eqs. 10 and 11 ten times, which was sufficient to reach convergence and obtain the solutions \( \hat{A} \) and \( \hat{B} \).

3.2. Linear mappings by regularized decomposition

With our second method, we search two 179-by-179 matrices \( A \) and \( B \) that minimize \( f \):

\[ f(A, B) = \sum_i ||Ap_i + Bq_i - g_i||^2 \]  

here \( p_i \) and \( q_i \) denote the two input streams and \( g_i \) the (golden) reference stream (\( i \) represents the frame index); all \( g_i \) are probability distributions with only the single component representing the correct state of that frame equal to 1.

Elementary algebra shows that the derivatives \( \nabla_A(f) \) and \( \nabla_B(f) \) are given by

\[ \nabla_A(f) = 2 \sum_i (Ap_i + Bq_i - g_i) p_i \]  
\[ \nabla_B(f) = 2 \sum_i (Bq_i + Ap_i - g_i) q_i \]  

Requiring the resulting stream to consist of valid probability vectors leads to the additional constraint that the row sums of \( A + B \) are one:

\[ h(A, B) = ||Ae + Be - e||^2 = 0 \]  

in which \( e \) denotes a 179-dim column vector consisting of 1’s.

Finally, using the Lagrange multipliers \( (\lambda_1, \lambda_2) \), the minimization of \( f \) in eq. 6 under the constraint \( h = 0 \) (using shorthand notations \( PQ^t = \sum_i p_i q_i^t \) etc.) leads to:

\[ A \cdot P^t + B \cdot Q^t - QP^t = \lambda_1 (A + B - 1) ee^t \]  
\[ A \cdot PQ^t + B \cdot QQ^t - QP^t = \lambda_2 (A + B - 1) ee^t \]  

The difference between lag 1 and 2 is small: apparently the merged stream replicates the golden reference stream at the frame level. The method, however, is unable to precisely predict the stream weights; only tendencies are clearly visible.

### 3.2.1. Results

Results are shown in Figures 2 and 3. Fig. 2 shows the proportion of frames of the re-estimated stream \( \hat{A}p_{\alpha} + \hat{B}p_{\alpha} \) of which the most likely state corresponds to that of the golden stream \( p_g \). Because the digits were modeled as a sequence of 16 HMM states, acoustic characteristics of neighboring states can be very similar. Moreover, small temporal differences in the output of the aligner are not likely to have a noticeable effect on the result of a Viterbi decoding. To account for this, we used three different correctness measures making use of the notion of 'lag' between states: the lag between two states is defined as the minimal distance along any eligible Viterbi path. For example, two states have lag 1 if they are neighbor states within a word, or e.g. when one state is word final, while the other state is word initial. For quantifying the match between merged stream and golden stream, we applied three definitions of correctness: The winning state is assumed correct if (1) lag = 0; it exactly matches the golden reference state, (2) lag ≤ 1; if it is the same as, or is neighbor of the reference state along a Viterbi path, and (3) lag ≤ 2: if it is the same as, or is the neighbor of, or the neighbor’s neighbor of the golden state.

Fig. 2 shows the match between the reference stream on the one hand, and the original MLP and SC streams and the merged stream on the other, as a function of SNR. Subsequent panels depict the results for lags 0, 1, and 2, respectively. \( \hat{A} \) and \( \hat{B} \) were estimated using the different SNR conditions of the training set. The figure convincingly shows that the merged stream better approximates the golden reference than each of the individual streams. Thus, also this second method shows that a proper merge can improve the frame-based trustworthiness. Furthermore, it supports our earlier finding that an analytical approach can lead to results that are very similar to those obtained via a laborious grid search as in [22].

The difference between lag 1 and 2 is small: apparently the remaining errors are long-distance errors, which are probably very hard to repair using a frame-based stream merging approach. A lag-2 correctness of about 90 percent is likely to be the best we can achieve.

Finally, we checked whether the properties of the found matrices \( \hat{A} \) and \( \hat{B} \) varied systematically as a function of SNR. From eq. 6, it follows that \( \hat{A} \) and \( \hat{B} \) jointly attempt to optimally map two observed streams onto a reference stream; in other words, \( \hat{A} \) and \( \hat{B} \) are in principle able to repair consistent state-to-state mislabelling errors (compared to the reference labelling) made by the individual classifiers. One of the methods to compare \( \hat{A} = (\hat{a}_{ij}) \) and \( \hat{B} = (\hat{b}_{ij}) \) and to investigate their behaviour across SNR is by using their ’energy’, i.e. the sum of the squared components \( \sum_i \hat{a}_{ij}^2 \) and \( \sum_i \hat{b}_{ij}^2 \). For a matrix this value is equal to the sum of the squared eigenvalues, and therefore a measure for the amount of explained variance.
to 8% (SNR -5) compared to the plain grid search approach. In summary, with increasing noise levels, it becomes increasingly more important.

The analytical solution was tested using the same Viterbi back end decoding system. The obtained recognition results were very comparable to those obtained using the extensive plain grid search optimization of stream weights as in [22]; the WERs showed a promising relative decrease of 5% (SNR 20) to 8% (SNR -5) compared to the plain grid search approach.

In [25, 17] it was shown that the optimal weights of streams are related to their inverse entropy $1/H$. In general, however, one does not know the relation between inverse entropy and the actual trustworthiness of a stream. When combining MLP and SC systems using the $1/H$-approach, one cannot assume that the inverse entropy is a measure for the stream trustworthiness that equally applies to both systems. Therefore, instead of considering $1/H$ as a measure for the trustworthiness of a stream, it seems beneficial to determine trustworthiness based on the actual accuracy of probability vectors, and to find a system-independent mapping from a posterior vector to an empirical trustworthiness value. The method in this paper provides a powerful step in this direction.

4. Discussion and conclusion

The weights required for optimizing word accuracy in a dual stream ASR system were empirically shown to depend on SNR in an systematic and monotonic way (see WERs in Fig. 1 and [22]). Although these observations seem quite logical given the SNRs, it is difficult to study which stream properties contribute to the success of a merged stream. The analytical tools for a mathematically principled merge scheme as presented in this paper is a substantial step forward. Using the assumption that equally applies to both systems. Therefore, instead of considering $1/H$ as a measure for the trustworthiness of a stream, it seems beneficial to determine trustworthiness based on the actual accuracy of probability vectors, and to find a system-independent mapping from a posterior vector to an empirical trustworthiness value. The method in this paper provides a powerful step in this direction.

4.1. Future work

Analytical approaches such as presented in this paper provide valuable insights in how a second stream can help to correct errors that would have been made by a single stream approach. We currently study how the analytical solution can serve as starting point for subsequent fine-grained grid-based optimizations. We also investigate whether the proposed methods can be extended to an analysis of the post back-end performance by means of analysis of the competition between the Viterbi decoding paths. Another future issue is the extension towards larger vocabulary recognition tasks and to more complex speech data.
5. References


