Uncertainty Decoding in Automatic Speech Recognition

Reinhold Häb-Umbach

Fachgebiet Nachrichtentechnik, Universität Paderborn, 33098 Paderborn
E-Mail: {haeb}@nt.uni-paderborn.de
Web: www-nt.uni-paderborn.de

Abstract

The term *uncertainty decoding* has been phrased for a class of robustness enhancing algorithms in automatic speech recognition that replace point estimates and plug-in rules by posterior densities and optimal decision rules. While uncertainty can be incorporated in the model domain, in the feature domain, or even in both, we concentrate here on feature domain approaches as they tend to be computationally less demanding. We derive optimal decision rules in the presence of uncertain observations and discuss simplifications which result in computationally efficient realizations. The usefulness of the presented statistical framework is then exemplified for two types of real-world problems: The first is improving the robustness of speech recognition towards incomplete or corrupted feature vectors due to a lossy communication link between the speech capturing front end and the backend recognition engine. And the second is the well-known and extensively studied issue of improving the robustness of the recognizer towards environmental noise.

1 Introduction

Improving the robustness of state-of-the-art automatic speech recognition (ASR) continues to be an important research area. Current hidden Markov model (HMM)-based speech recognition systems are notorious for performing well in matched training and test conditions while quickly degrading in the presence of a mismatch. While such a mismatch may be caused by many factors, probably one of the most studied problems is improving the robustness of a recognizer trained on clean training data to test data being corrupted by environmental noise.

Huge research efforts have been devoted to overcoming this lack of robustness, and a wealth of methods has been proposed. These can be categorized into methods that either try to compensate the effect of distortions on the features (so-called *front-end* methods) or approaches that modify the models used in the recognizer to better match the incoming distorted feature stream (*back-end* methods).

Traditionally, front-end methods aim at obtaining point estimates of the uncorrupted, clean features. Likewise, back-end methods usually try to obtain point estimates of parameters, such as the mean vectors of the observation probabilities. These estimates are then "plugged" into the Bayesian decision rule as if they were perfect estimates. However, more recently the focus has shifted to estimating the features or parameters together with a measure of reliability of the estimate and propagating the uncertainty to the decision rule [1,7,10,5,17,19,21,23,26,27]. The underlying rationale is that an estimate is never perfect and that the recognizer can benefit from knowing the estimation error variance by deemphasizing the contributions of unreliable estimates to the overall decision on the word sequence.

The use of such optimal decision rules is by no means new. How to modify the Bayesian decision rule in the presence of missing or noisy features can be found in many textbooks on pattern recognition, see e.g. [12]. Two developments, however, are fairly recent: how to modifiy the decision rule for HMM classifiers and how to obtain reliability information for a given class of distortions.

In this paper we first rederive in a slightly different manner an optimal decision rule for HMM-based speech recognition in the presence of corrupted feature vectors, which we have originally presented in [21]. We specifically place emphasis on exploiting the temporal correlation among the feature vectors, thus relaxing – to some extent – the conditional independence assumption, which is generally used in HMM based speech recognition and which is considered to be one of its major shortcomings. If appropriate approximations are made, the new decision rule requires only an adjustment of the likelihood computation compared to the classical decoder.

A key element of this new decision rule is the posterior density of the clean feature vector, given the observed and corrupted feature vectors. We show how this posterior can be estimated for two types of signal degradations: First, for a remote ASR system, where the speech capturing unit is connected to the decoder via an error-prone communication network, and second, for feature vectors degraded by additive environmental noise.

This paper is organized as follows: In the next section we formulate the speech recognition optimization problem in the presence of uncertain features and correlation among successive feature vectors. From this an uncertainty decoding rule is derived and it is shown that other decoding rules found in the literature can be obtained as special cases of this more general rule. After discussing realization issues in Section 3 we discuss the feature posterior estimation for the two classes of signal degradation mentioned above. Section 5 presents experimental results comparing uncertainty decoding with plug-in rules.

2 Optimal Decoding Rules

Starting from the classical Bayesian framework for speech recognition, we subsequently extend it to account for corrupted observations and correlation among successive feature vectors.

2.1 Bayesian Framework of Speech Recognition

Given a sequence of feature vectors $\mathbf{x}_1^T = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ of length T extracted from an utterance, the classification task comes down to finding that sequence of words $\hat{\mathbf{W}}$ from a given vocabulary which maximizes the joint probability $p(\mathbf{W}, \mathbf{x}_1^T)$ or, equivalently,

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}\, p(\mathbf{x}_1^T|\mathbf{W}) \cdot P(\mathbf{W}). \quad (1)$$

The a priori probability of the word sequence, $P(\mathbf{W})$, is provided by the language model, while the acoustic model

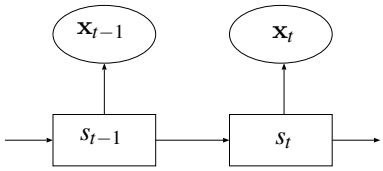

Figure 1: Bayesian network prevalent in speech recognition.

is concerned with computing $p(\mathbf{x}_1^T|\mathbf{W})$. In a HMM-based speech recognizer this is accomplished by introducing the sequence of hidden states $s_1^T = (s_1, \ldots, s_T)$ underlying the sequence of observations:

$$p(\mathbf{x}_1^T|\mathbf{W}) = \sum_{\{s_1^T\}} p(\mathbf{x}_1^T|s_1^T) \cdot P(s_1^T|\mathbf{W}) \qquad (2)$$

where the summation is carried out over all state sequences within $\mathbf{W}$. In order to solve (2) recursively, both terms under the sum are factorized:

$$p(\mathbf{x}_1^T|s_1^T) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_1^{t-1}, s_1^T) \qquad (3)$$

$$P(s_1^T|\mathbf{W}) = \prod_{t=1}^{T} P(s_t|s_1^{t-1}, \mathbf{W}). \qquad (4)$$

These expressions can be simplified by considering the statistical dependencies among the random variables as depicted by the Bayesian network of Fig. 1. This figure illustrates the assumptions made in most of today's speech recognition engines: the state sequence being a first-order Markov process and the feature vectors being conditionally independent:

$$P(s_1^T|\mathbf{W}) = \prod_{t=1}^{T} P(s_t|s_{t-1}, \mathbf{W}) \qquad (5)$$

$$p(\mathbf{x}_1^T|s_1^T) = \prod_{t=1}^{T} p(\mathbf{x}_t|s_t). \qquad (6)$$

Using (5) and (6) in (2) we arrive at the well known result

$$p(\mathbf{x}_1^T|\mathbf{W}) = \sum_{\{s_1^T\}} \prod_{t=1}^{T} p(\mathbf{x}_t|s_t) \cdot P(s_t|s_{t-1}, \mathbf{W}). \qquad (7)$$

An approximate value for (7) can be computed by the Viterbi algorithm.

2.2 Presence of Corrupted Features

In the following we denote $\mathbf{x}_1^T$ the sequence of "clean" or uncorrupted features (assuming that training has been carried out with uncorrupted data). In many practical cases there exists a mismatch between training and testing conditions. This can be expressed by the fact that the sequence of test features $\mathbf{x}_1^T$, which are representative of the training conditions, is not observable. Instead of $\mathbf{x}_1^T$, a corrupted version $\mathbf{y}_1^T$ is observed. The latter may differ from the former due to environmental noise or due to errors occurring during transmission over a communication network, e.g. in a distributed speech recognition setup.

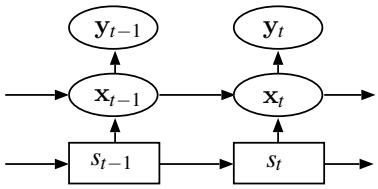

Figure 2: Bayesian network considering temporal correlation between features.

As $\mathbf{x}_1^T$ is not available, the recognition task is stated now as finding the word sequence most likely to yield $\mathbf{y}_1^T$:

$$\hat{\mathbf{W}} = \operatorname*{argmax}_{\mathbf{W}} p(\mathbf{y}_1^T|\mathbf{W}) \cdot P(\mathbf{W}). \qquad (8)$$

Taking $\mathbf{y}_1^T$ as if they were the "clean", uncorrupted data, i.e. interpreting $\mathbf{y}_1^T$ as an estimate of $\mathbf{x}_1^T$ to be used in (1) results in the well-known poor performance of speech recognition in the presence of a mismatch between training and testing conditions.

In a "plug-in" decision rule we would replace the observed $\mathbf{y}_1^T$ by estimates $\hat{\mathbf{x}}_1^T$ of the clean feature vectors obtained from $\mathbf{y}_1^T$ and leave (8) or (1) otherwise unchanged.

However, one can do better by accounting for the unreliability in these estimates. To this end we introduce the unobservable (hidden) sequence $\mathbf{x}_1^T$ of clean speech feature vectors:

$$p(\mathbf{y}_1^T|\mathbf{W}) = \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) p(\mathbf{x}_1^T|\mathbf{W}) d\mathbf{x}_1^T, \qquad (9)$$

where the notation $\{\mathbf{x}_1^T\}$ shall indicate that the integration has to be carried out over all possible feature vector sequences of length T.

The Bayesian network of Fig. 2 depicts the assumed statistical dependencies among the random variables under consideration. Note that the observed feature vectors are statistically independent of the HMM states, if the clean features are given; and further note that we assumed a direct statistical dependency among subsequent clean feature vectors, thus abandoning the conditional independence assumption.

Again introducing the HMM state sequence, the acoustic search now has to compute:

$$p(\mathbf{y}_1^T|s_1^T) = \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) p(\mathbf{x}_1^T|s_1^T) d\mathbf{x}_1^T, \qquad (10)$$

where

$$p(\mathbf{x}_1^T|s_1^T) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_1^T). \qquad (11)$$

Using (11) in (10) we obtain

$$\begin{aligned} p(\mathbf{y}_1^T|s_1^T) &= \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_1^T) d\mathbf{x}_1^T \\ &\propto \int_{\{\mathbf{x}_1^T\}} \frac{p(\mathbf{x}_1^T|\mathbf{y}_1^T)}{p(\mathbf{x}_1^T)} \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_1^T) d\mathbf{x}_1^T \\ &= \int_{\{\mathbf{x}_1^T\}} \prod_{t=1}^{T} \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_1^T)}{p(\mathbf{x}_t|\mathbf{x}_{t-1})} p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_1^T) d\mathbf{x}_1^T \end{aligned} \qquad (12)$$

To be able to further simplify (12) we first assume that $\prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_1^T) \approx \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_t)$, which is justified by the fact that the dependency between $\mathbf{x}_t$ and $(s_t, \mathbf{x}_{t-1})$ is stronger than between $\mathbf{x}_t$ and s_{t+1}, $s_{t+2}, \ldots$. Disregarding also the dependency on $\mathbf{x}_{t-1}$ allows us to change the order of integral and product:

$$\begin{aligned} p(\mathbf{y}_1^T|s_1^T) &\approx \int_{\{\mathbf{x}_1^T\}} \prod_{t=1}^{T} \frac{p(\mathbf{x}_t|\mathbf{y}_1^T)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_1^T \\ &= \prod_{t=1}^{T} \int_{\{\mathbf{x}_t\}} \frac{p(\mathbf{x}_t|\mathbf{y}_1^T)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t, \quad (13) \end{aligned}$$

and only after this change we are still able to carry out decoding using the Viterbi algorithm or any other search technique established in speech recognition.

A comparison of (13) with (6) reveals that the only difference to the classical decoding rule presented in Section 2.1 is that the observation likelihood $p(\mathbf{x}_t|s_t)$ has to be replaced by the likelihood

$$p_{LH}(\mathbf{y}_1^T|s_t) = \int_{\{\mathbf{x}_t\}} \frac{p(\mathbf{x}_t|\mathbf{y}_1^T)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t \quad (14)$$

This decoding rule has been first published in [21], however here we have presented a sightly different derivation.

2.3 Related Uncertainty Decoding Rules

One might wonder whether the derived decoding rule really accounts for direct inter-frame correlation of feature vectors or whether we are not just back with the conditional independence assumption, since we had to disregard the dependencies on $\mathbf{x}_{t-1}$ in (12) to arrive at an optimization problem tractable by standard speech recognition engines. A closer look at the derivation reveals that the relaxation of the conditional independence assumption is reflected by the posterior $p(\mathbf{x}_t|\mathbf{y}_1^T)$ being conditioned on all observed feature vectors. If conditional independence were assumed, i.e. if $\mathbf{x}_t$ were related to $\mathbf{x}_{t-1}$ only indirectly via s_t and s_{t-1}, then

$$\begin{aligned} p(\mathbf{y}_1^T|\mathbf{x}_1^T) p(\mathbf{x}_1^T|s_1^T) &= \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|s_t) \quad (15) \\ &\propto \prod_{t=1}^{T} \frac{p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t), \quad (16) \end{aligned}$$

i.e. the likelihood would now read

$$p_{LH}^{(CI)}(\mathbf{y}_t|s_t) = \int_{\{\mathbf{x}_t\}} \frac{p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t \quad (17)$$

where the superscript $(\cdot)^{(CI)}$ shall indicate conditional independence.

This simplified version of uncertainty decoding has been proposed earlier in the framework of noise-robust speech recognition [10, 4, 22, 23, 24]. The variants found in the literature differ in the way the posterior $p(\mathbf{x}_t|\mathbf{y}_t)$ or, alternatively, the joint density $p(\mathbf{x}_t, \mathbf{y}_t)$ is obtained.

In [7] and [27] the denominator $p(\mathbf{x}_t)$ has been neglected – an approximation that has not been identified as such. This approximation can be motivated on the grounds that the prior $p(\mathbf{x}_t)$ should have a larger variance than the posterior. Thus the denominator can be considered constant for the range of values of $\mathbf{x}_t$, where the posterior assumes values significantly larger than zero. However, this argument does no longer hold in the presence of strong distortions, e.g. low SNR. Then the use of the approximate decision rule, which neglects the prior, results in artifacts and poor performance. This inconsistency has been observed in [23], where, however, the reason for it remained unclear.

The potential superiority of the likelihood (14) over (17) can be illustrated with the following example: Consider a distortion which makes the observation at time t completely unreliable and leaves all other observations unaffected. Using (17), the frame at time t is marginalized. On the other hand, (13) is able to exploit the correlation in the sequence of feature vectors since the posterior is computed taking into account not only the instantaneous observed value $\mathbf{y}_t$ but also past and future observations. Thus, an uninformative or lost observation at time t no longer results in a constant observation probability at time t. Discrimination is still possible as long as the feature posterior $p(\mathbf{x}_t|\mathbf{y}_1^T)$ does not equal the prior $p(\mathbf{x}_t)$. That actually means that some part of the lost information can be recovered from reliable neighboring features.

3 Realization Issues

The good news is the that the search architecture need not be changed for uncertainty decoding. Only the evaluation of the observation likelihood has to be modified. However, the bad news is that a numerical evaluation of the integral as required by the modified observation probability would increase the computational burden beyond the limits of practical interest. Fortunately, the integral can be solved analytically by making the following assumptions:

1. The state conditioned observation probability of the uncorrupted feature is a Gaussian mixture:

$$p(\mathbf{x}_t|s_t) = \sum_{m=1}^{M} c_{s_t,m} \mathcal{N}(\mathbf{x}_t; \mu_{s_t,m}, \Sigma_{s_t,m}) \quad (18)$$

   where $c_{s_t,m}$ is the weight, $\mu_{s_t,m}$ the mean vector and $\Sigma_{s_t,m}$ the covariance matrix of the m-th mixture component of the observation probability of state s_t. This is the common assumption in speech recognition.

2. The a priori probability density of the clean speech feature is a Gaussian:

$$p(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}). \quad (19)$$

   Experimental data have proven this assumption to be quite valid, with certain reservations concerning the log energy component. Its probability density function has multiple peaks making single Gaussian modeling a rather coarse approximation.

3. The feature posterior given the observations is a Gaussian:

$$p(\mathbf{x}_t|\mathbf{y}) = \mathcal{N}(\mathbf{x}_t; \mu_{\mathbf{x}_t|\mathbf{y}}, \Sigma_{\mathbf{x}_t|\mathbf{y}}). \quad (20)$$

   Here the notation $\mathbf{x}_t|\mathbf{y}$ stands for either $\mathbf{x}_t|\mathbf{y}_t$ or $\mathbf{x}_t|\mathbf{y}_1^T$. Eq. (20) is the most debatable assumption, as we often observed a multi-modal shape of the posterior. Some researchers therefore suggested to use a Gaussian mixture model instead [27]. As this, however, has a strong impact on the computational effort, we prefer to stick to the model of a single Gaussian here.

If we assume all Gaussians of Eqs. (18-20) to have diagonal covariance matrices, the observation probability of Eqs. (14) and (17) can be factorized over the feature vector dimensions. Thus, we obtain:

$$\begin{aligned}&\sum_{m=1}^{M} c_{s_t,m} \int_{\{x_t\}} \mathcal{N}(x_t;\mu_{s_t,m},\sigma^2_{s_t,m}) \\ &\quad \cdot \frac{\mathcal{N}(x_t;\mu_{x_t|\mathbf{y}},\sigma^2_{x_t|\mathbf{y}})}{\mathcal{N}(x_t;\mu_x,\sigma^2_x)} dx_t \qquad (21)\\ &= \sum_{m=1}^{M} c'_{s_t,m} \mathcal{N}(\mu_{e_t};\mu_{s_t,m},\sigma^2_{s_t,m}+\sigma^2_{e_t}).\end{aligned}$$

where the equivalent mean μ_{e_t}, variance $\sigma^2_{e_t}$ and weights $c'_{s_t,m}$ are given by the following equations [12]:

$$\frac{\mu_{e_t}}{\sigma^2_{e_t}} = \frac{\mu_{x_t|\mathbf{y}}}{\sigma^2_{x_t|\mathbf{y}}} - \frac{\mu_x}{\sigma^2_x} \qquad (22)$$

$$\frac{1}{\sigma^2_{e_t}} = \frac{1}{\sigma^2_{x_t|\mathbf{y}}} - \frac{1}{\sigma^2_x} \qquad (23)$$

$$c'_{s_t,m} = c_{s_t,m} \frac{\mathcal{N}(0;\mu_{x_t|\mathbf{y}},\sigma^2_{x_t|\mathbf{y}})}{\mathcal{N}(0;\mu_x,\sigma^2_x)\mathcal{N}(0;\mu_{e_t},\sigma^2_{e_t})}. \qquad (24)$$

Note, however, that from the assumption of a diagonal covariance of the prior and the observation likelihood it cannot be inferred that the posterior should also be Gaussian. This depends very much on the kind of distortion and notably for additive environmental noise the diagonal assumption for the posterior is quite coarse.

Eq. (21) states that the originally trained observation probability of the clean feature must be changed by increasing the variance by $\sigma^2_{e_t}$, and that it has to be evaluated at μ_{e_t}.

This variance has to be added to all acoustic model components, which totals a few hundreds for a small task but which can easily exceed hundred thousand for a large task, such as a Broadcast News system. Moreover, this variance addition is not as simple to apply as the Gaussian normalization term that is usually cached must be recomputed.

An unfavorable side effect of techniques which broaden the observation probability densities, such as uncertainty decoding, is that the search space increases due to reduced discriminability between the word hypotheses in the presence of uncertain observations. As the observation probability of an unreliable feature tends to be the same for all model states, the beam pruning looses efficiency and the number of "active" states increases. These factors lead to a slowdown of the recognition computation, the degree depending on the problem at hand.

4 Feature Posterior Estimation

The key element of the uncertainty decoding rule is the feature posterior $p(\mathbf{x}_t|\mathbf{y}_1^T)$, and the success of uncertainty decoding crucially depends on how well it can be determined.

Knowledge of the posterior density enables one to compute an optimal estimate with respect to any criterion. For example the minimum mean-square error (MMSE) estimate is the conditional mean $\hat{\mathbf{x}}^{\text{MMSE}} = \int \mathbf{x}_t p(\mathbf{x}_t|\mathbf{y}_1^T) d\mathbf{x}_t$. Similarly, a measure of accuracy of the estimate can be obtained form the posterior. In the Gaussian case the variance of the posterior is even identical with the estimation error variance.

Conceptually, the posterior can be estimated recursively via the following equations, where we have restricted ourselves to causal processing, i.e. rather than computing $p(\mathbf{x}_t|\mathbf{y}_1^T)$ we compute $p(\mathbf{x}_t|\mathbf{y}_1^t)$:

$$p(\mathbf{x}_t|\mathbf{y}_1^{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_1^{t-1}) d\mathbf{x}_{t-1} \qquad (25)$$

$$p(\mathbf{x}_t|\mathbf{y}_1^t) = \frac{p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_1^{t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_1^{t-1}) d\mathbf{x}_t} \qquad (26)$$

By using $p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_1^{t-1}) = p(\mathbf{y}_t|\mathbf{x}_t)$ in the last equation, we assumed independent and identically distributed observations.

For the determination of the posterior, the following issues have to be addressed:

- A dynamical model of the clean speech feature trajectory has to be established, delivering the term $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ needed in (25).
- An observation model $p(\mathbf{y}_t|\mathbf{x}_t)$ to be used in (26) has to be derived for the problem at hand.
- An appropriate inference algorithm has to be chosen or developed, which computes (an approximation of) $p(\mathbf{x}_t|\mathbf{y}_1^t)$ or $p(\mathbf{x}_t|\mathbf{y}_1^T)$. Note that in general the posterior cannot be determined analytically. The implementation may require the storage of the entire (non-Gaussian) pdf which is, in general terms, equivalent to an infinite dimensional vector [25]. Therefore one often has to resort to approximative, sub-optimal solutions.

In the following we discuss these three issues for the following two applications

- Distributed Speech Recognition (DSR), where quantized features are transmitted over a lossy communication link.
- Speech corrupted by additive environmental noise.

4.1 Feature Posterior Estimation for DSR

In Distributed Speech Recognition (DSR) a client carries out feature extracion and transmits the coded features via a communication network to a server, which conducts the actual speech recognition. To ensure the interoperability of various equipments, the ETSI-DSR standards were developed by the Aurora working group [13, 14]. They define the feature extraction and quantization algorithms to be used in a front end of a DSR system. Fig. 3 shows a block diagram of the DSR system featuring the elements relevant for feature posterior computation.

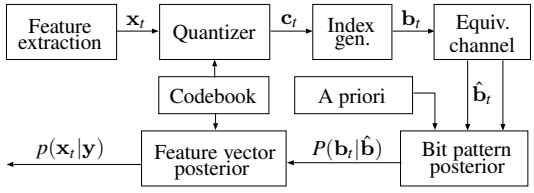

Figure 3: Block diagram of processing elements relevant for feature posterior estimation in a DSR system.

As can be seen in Fig. 3 the features are quantized to centroids $\mathbf{c}_t$ and mapped to bit patterns $\mathbf{b}_t$ which are

transmitted over an error-prone equivalent discrete channel. The correlation among bit patterns successive in time is modeled as a first-order Markov process. Note that the transmitted bit patterns $\mathbf{b}_t$ and the received $\hat{\mathbf{b}}_t$ are discrete random variables. The dynamics of the clean and quantized features can therefore be described by the a priori probability matrix $P(\mathbf{b}_t^{(i)}|\mathbf{b}_{t-1}^{(j)})$, where $\mathbf{b}_t^{(i)}$ denotes the i-th out of 2^M bit patterns of length M; i.e. $i,j \in \{1,\ldots,2^M\}$. They can be estimated on clean speech training data.

The effect of the channel is captured by the observation likelihood $P(\hat{\mathbf{b}}_t|\mathbf{b}_t)$. In [21] we have shown how this term can be computed for channels exhibiting bit errors or packet losses.

An appropriate inference algorithm in the presence of discrete Markov sources is the Forward-Backward algorithm [2] which computes

$$\gamma_t(i) = P(\mathbf{b}_t = \mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{2^M}\alpha_t(j)\beta_t(j)} \qquad (27)$$

by a recursion on the forward probabilities $\alpha_t(i)$ and a recursion on the backward probabilities $\beta_t(i)$, defined by:

$$\alpha_t(i) = P(\hat{\mathbf{b}}_1^t, \mathbf{b}_t = \mathbf{b}_t^{(i)}) \qquad (28)$$

$$\beta_t(i) = P(\hat{\mathbf{b}}_{t+1}^T|\mathbf{b}_t = \mathbf{b}_t^{(i)}). \qquad (29)$$

Once the discrete bit pattern posterior $P(\mathbf{b}_t = \mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T)$ is obtained, the continuous feature posterior can be derived from it [20].

4.2 Feature Posterior Estimation for noise-robust ASR

The effect of additive environmental noise on the cepstral feature vectors is highly non-linear. In a simplified model which neglects the inner product between the clean speech and noise, the noisy cepstral feature vector is given by

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{M}_{DCT}\log\left(1 + e^{\mathbf{M}_{DCT}^+(\mathbf{n}_t - \mathbf{x}_t)}\right), \qquad (30)$$

where $\mathbf{x}_t$ and $\mathbf{n}_t$ denote the clean speech and noise-only ceptral feature vectors. Here, $\mathbf{y}_t$, $\mathbf{x}_t$ and $\mathbf{n}_t$ are continuous random variables. $\mathbf{M}_{DCT}$ and $\mathbf{M}_{DCT}^+$ are the Discrete Cosine Transform matrix and its pseudo-inverse, respectively.

Switching Linear Dynamical Models (SLDM) have been proposed to model the dynamics of the clean speech feature trajectory [9], since a single linear dynamical model cannot account sufficiently well for the complicated dynamics of speech. In a SLDM $\mathbf{x}_t$ is described by a piecewise linear dynamical model, where a discrete regime variable determines which LDM is active at a time:

$$\begin{aligned}\mathbf{x}_t &= \mathbf{A}(\theta_t)\mathbf{x}_{t-1} + \mathbf{b}(\theta_t) + \mathbf{u}_t \\ \mathbf{u}_t &\sim \mathcal{N}(\mathbf{u}_t; 0, \mathbf{C}(\theta_t))\end{aligned} \qquad (31)$$

Here, the state transition matrix $\mathbf{A}(\theta_t)$, the bias $\mathbf{b}(\theta_t)$ and the covariance matrix of the system noise $\mathbf{C}(\theta_t)$ depend on the regime variable $\theta_t \in \{1,\ldots,M_\theta\}$, where M_θ is the number of dynamical models. The parameters can be estimated on clean speech training data by a variant of the Expectation-Maximization (EM) algorithm [9].

Exact inference for a SLDM is computationally intractable as the complexity increases exponentially with time, since every possible history of the regime variable has to be considered. A number of approximate inference algorithms has been proposed, such as the Generalized Pseudo-Bayes (GPB) and the Interacting Multiple Model (IMM) algorithm [3], where the Gaussian mixture resulting after each iteration is collapsed to a single Gaussian before advancing to the next iteration.

Another issue is the non-linear observation model (30). It is either statistically or analytically linearized resulting in a bank of unscented or extended Kalman filters for posterior estimation.

5 Experimental Results

In this section we present experimental results on the Aurora 2 task [18] . It consists of 4004 utterances (continuously spoken digit strings) from 52 male and 52 female speakers, distributed over four subsets. The sampling rate is 8kHz. There is no language model for this task. The acoustic models have been trained in clean conditions as described in [18]. With feature vectors computed using the ETSI DSR standard [14] the WER in error-free conditions (i.e. without channel-induced errors or additive noise) is 0.86%.

For the experiments on DSR we used this noise-free test set, while for the experiments on noise robustness we employed test set A, which consists of four subsets of 1001 utterances each. To each subset noise of a different type has been artificially added. These noise types are subway, babble, car and exhibition.

5.1 Packet Loss

In this section we consider Distributed Speech Recognition over an IP channel, where packet erasure is the dominant error pattern. This phenomenon can be attributed to network congestions but also to possible bit errors in the low-level network layer which alter the packet checksum. Often, the 2-state Gilbert model [15] is used to model the bursty nature of packet losses. The model has two parameters: the mean loss probability mlp, which is the average probability of packet loss, and the conditional loss probability clp, which is the probability of packet loss given that the previous packet was also lost. The parameters depend on many factors such as network load, packet size, etc., however, for simulation purposes some authors including [6, 8, 16] extensively used the settings given in Table 1.

Table 1: The conditional loss probability (*clp*) and mean loss probability (*mlp*) for the simulated network conditions.

Condition	C1	C2	C3	C4
clp	0.147	0.33	0.5	0.6
mlp	0.006	0.09	0.286	0.385

The number of feature vectors that can be accommodated in a payload may vary, however in our experiments we used one frame-pair (two feature vectors) per packet.

Table 2 presents the word accuracies obtained on Aurora 2 for the following variants of the decoding rule. For "MMSE0" and "MMSE1", the posteriors are collapsed to their means: $p(\mathbf{x}_t|\mathbf{y}) \approx \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t)$, where $\hat{\mathbf{x}}_t = E[\mathbf{x}_t|\mathbf{y}]$.

Here, the notation $\mathbf{x}_t|\mathbf{y}$ stands for $\mathbf{x}_t|\mathbf{y}_t$ in the case of "MMSE0" and $\mathbf{x}_t|\mathbf{y}_1^T$ in the case of "MMSE1". Collapsing the posterior to a point estimate results in the well-known plug-in rules: the estimates are taken as if they were perfect, i.e. the likelihood $p(\hat{\mathbf{x}}_t|s_t)$ is used in the recognizer.

The entries in the rows "UD0" and "UD1" refer to uncertainty decoding, where the likelihood computation was carried out according to (17) and (14), respectively. Further the results obtained with the error concealment method proposed in the ETSI standard (nearest frame repetition) is given as a baseline.

It can be clearly observed that uncertainty decoding improves performance over plug-in rules. Note that in this setup the use of decoding rule (17) (UD0) actually corresponds to marginalization in case of a lost feature. Another consequence is that MMSE0 reduces to simply inserting the feature a priori mean value in the gap periods, resulting in very poor performance.

Table 2: Word accuracy [%] on the Aurora 2 task for transmission over the packet-switched network with 2 features vectors per packet.

Approach	C1	C2	C3	C4
ETSI (NFR)	99.10	98.94	97.33	94.80
MMSE0	98.33	86.59	60.37	49.02
MMSE1	99.10	98.97	97.58	94.26
UD0	99.07	98.81	97.11	94.63
UD1	99.10	99.02	98.34	96.91

5.2 Environmental Noise

In this subsection we experiment with uncertainty decoding as a means to improve the robustness of the recognizer to environmental noise. Table 3 shows the results for the subsets of test set A averaged over SNRs 0, 5, 10, 15 and 20 dB. The first two lines give the results obtained with the Standard ("SFE") and Advanced Front End ("AFE"), as standardized by ETSI [13, 14].

As outlined in section 4.2 a SLDM was used to model the dynamics of speech. For the experiments reported here, $M_\theta = 16$ different models were used. The Generalized Pseudo-Bayesian Algorithm was used to compute the posterior $p(\mathbf{x}_t|\mathbf{y}_1^t)$ leading to the results of the line "UD1c" (c: causal processing), while an additional backward filtering using a heuristics outlined in [11] leads to the posterior $p(\mathbf{x}_t|\mathbf{y}_1^T)$ (UD1). For the computation of the posterior $p(\mathbf{x}_t|\mathbf{y}_t)$ to be used for MMSE0 and UD0 decoding the state transition matrix $\mathbf{A}(\theta_t)$ in (31) is set to zero, resulting in a Gaussian mixture model for $p(\mathbf{x}_t)$.

The results show a similar trend as those of Table 2: plug-in decoding rules are inferior to uncertainty decoding and a clean speech posterior conditioned on all observations is superior to a posterior conditioned on the current and all past, which in turn is superior to a posterior conditioned only on the current observation.

It should also be noted that it is common practice to use a heuristics in uncertainty decoding for noise-robust speech recognition, for which no satisfying theoretical explanation has been given sofar. Both in [10], [23] and also for the results presented here, the variance of the posterior has been thresholded to make sure that it is sufficiently lower than the variance of the prior. We used the following rule: if $\sigma^2_{\mathbf{x}_t|\mathbf{y}} > 0.05\sigma^2_{\mathbf{x}}$, then $\sigma^2_{\mathbf{x}_t|\mathbf{y}} = 0.05\sigma^2_{\mathbf{x}}$.

Overall, the results obtained by posterior estimation and uncertainty decoding are somewhat disappointing, as the performance of the Advanced Front End could not be attained.

Table 3: Word accuracy [%] on test set A of the Aurora 2 task; averages taken over 0, 5, 10, 15 and 20 dB SNR.

Approach	Sub.	Bab.	Car	Exh.	Avg
SFE	68.06	44.74	59.97	68.73	60.37
AFE	88.83	84.82	90.82	88.69	88.29
MMSE0	76.90	72.93	79.68	76.07	76.40
MMSE1	80.19	72.56	84.28	82.43	79.87
UD0	78.74	75.96	81.71	76.75	78.29
UD1c	82.75	71.10	88.36	83.09	81.32
UD1	85.35	67.79	80.55	78.73	83.33

6 Conclusions

In this paper we have shown that uncertainty decoding is a powerful concept to improve the robustness of speech recognition towards a mismatch between training and testing conditions. The key to success is, however, the estimation of the posterior probability of the clean speech feature vector given the corrupted. We have shown for two types of distortions, packet and thus feature vector losses in the case of distributed speech recognition (DSR) and additive environmental noise, how the posterior can be estimated. While in the first case (DSR), uncertainty decoding greatly improved the immunity of the recognition engine towards lost feature vectors, in the second, current approaches hardly achieve the performance of the ETSI Advanced Front End. Indeed, the second seems to be the harder problem, one reason being that the effect of additive noise on cepstral features is highly non-linear. More research and, hopefully, insight is needed to better understand the shortcomings of today's approaches.

7 Acknowldegement

The author gratefully achknowledges the excellent work of Valentin Ion and Stefan Windmann who contributed much to the issues discussed in this paper.

References

[1] J.A Arrowood and M.A. Clements. Using observation uncertainty in HMM decoding. In *Proc. of ICSLP, Denver, Colorado*, 2002.

[2] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Information Theory*, 10:284–287, March 1974.

[3] Y. Bar-Shalom, X. Rong Li, and T. Krubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.

[4] J. Barker, M. Cooke, and D. Ellis. Decoding speech in the presence of other sources. *Speech Communication*, 45:5–25, 2005.

[5] A. Bernard and A. Alwan. Joint channel decoding - Viterbi recognition for wireless applications. In *Proc. of EUROSPEECH, Aalborg, Denmark*, 2001.

[6] C. Boulis, M. Ostendorf, E:A. Riskin, and S. Otterson. Graceful degradation of speech recognition performance over packet-erasure networks. *IEEE Trans. Speech and Audio Processing*, 10(8):580–590, Nov. 2002.

[7] L. Deng, J. Droppo, and A. Acero. Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. *IEEE Trans. Speech and Audio Processing*, 13(3):412–421, 2005.

[8] L. Docío-Fernández and C. García-Mateo. Distributed speech recognition over IP networks on the Aurora 3 database. In *Proc. of ICSLP, Denver, USA*, 2002.

[9] J. Droppo and A. Acero. Noise robust speech recognition with a switching linear dynamic model. In *Proc. of ICASSP, Montreal*, 2004.

[10] J. Droppo, A. Acero, and L. Deng. Uncertainty decoding with SPLICE for noise robust speech recognition. In *Proc. of ICASSP, Orlando, Florida*, 2002.

[11] J. Droppo, L. Deng, and A. Acero. A comparison of three non-linear observation models for noisy speech features. In *Proc. of Eurospeech, Geneva, Switzerland*, 2003.

[12] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[13] ES.201.108. Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms. *ETSI*, April 2000.

[14] ES.202.050. Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms. *ETSI*, Oct 2002.

[15] E.N. Gilbert. Capacity of a burst-noise channel. *The Bell System Technical Journal*, Sep. 1960.

[16] A.M. Gómez, A.M. Peinado, V. Sánchez, and A.J. Rubio. A source model mitigation technique for distributed speech recognition over lossy packet channels. In *Proc. of EUROSPEECH*, 2003.

[17] R. Haeb-Umbach and V. Ion. Soft features for improved distributed speech recognition over wireless networks. In *Proc. of ICSLP, Jeju, Korea*, 2004.

[18] H. Hirsch and D. Pearce. The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ISCA ITRW Workshop ASR2000, Paris, France*, 2000.

[19] Q. Huo and C-H. Lee. A Bayesian predictive approach to robust speech recognition. *IEEE Trans. Speech and Audio Processing*, 8(8):200–204, 2000.

[20] V. Ion and R. Haeb-Umbach. Uncertainty decoding for distributed speech recognition over error-prone networks. *Speech Comm.*, 48:1435–1446, 2006.

[21] V. Ion and R. Haeb-Umbach. A novel uncertainty decoding rule with applications to transmission error robust speech recognition. *IEEE Trans. on Audio, Speech, and Language Processing*, 16:1047–1060, 2008.

[22] T. T. Kristjansson and B. J. Frey. Accounting for uncertainty in observations: A new paradigm for robust automatic speech recognition. In *Proc. of ICASSP*, 2002.

[23] H. Liao and M. J. F. Gales. Issues with uncertainty decoding for noise robust speech recognition. *Speech Communication*, 50:265–277, 2008.

[24] Andrew Morris, Jon Barker, and Herv Bourlard. From missing data to maybe useful data: soft data modelling for noise robust ASR. *Proc. WISP*, 06, 2001.

[25] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter – Particle Filters for Tracking Applications*. Artech House, 2004.

[26] V. Stouten, H. van Hamme, and P. Wambacq. Accounting for the uncertainty of speech estimates in the context of model-based feature enhancement. In *Proc. of ICSLP, Jeju Island, Korea, Oct. 2004*.

[27] V. Stouten, H. van Hamme, and P. Wambacq. Model-based feature enhancement with uncertainty decoding for noise robust ASR. *Speech Communication*, 48, Issue 11, Nov. 2006.