Optimal Detection in Case of the Sparse Training Data

David Klusáček

ÚFAL, Faculty of Mathematics and Physics,
Charles University, Prague
<klusacek@atrey.karlin.mff.cuni.cz>

ODYSSEY04 - The Speaker and Language Recognition Workshop
Toledo, Spain
May 31 - June 3, 2004

Abstract

We will treat a task of statistical detection using discrete alphabet in this paper. It is well known that the likelihood ratio detector is the optimal one, provided that "measurements" taken on the detected object are mutually independent and that we know the feature distributions of target and background objects precisely. The detector presented here is optimal using only the independence assumption. Instead of requiring the knowledge of the underlying distributions it relies on the training data itself. Further, we introduce the averaging technique which aims to lower the effects of statistical dependence. This averaged detector outperformed similarly averaged likelihood ratio detector by 7% relative in the task of speaker detection.

1. Introduction

The detection task using the discrete alphabet has to do the following job: We have some objects divided into two classes — *target class* and *background class*. The objects exhibit some behavior which can be measured and represented as a symbol of some suitable finite alphabet. Depending on circumstances we can have more measurements belonging to a single object. We will treat them as if they were statistically mutually independent (although it is rarely the case in the real life). For instance, think of the objects as of speakers, of the result of the measurement as of the phone the speaker pronounces during a given 1/100 second and of "more measurements" as of the collection of single measurements gathered thru the conversation (they will be dependent, so in fact it is not very good example). Now, we would like to have an algorithm which will decide, according to the collection of measurements, which class the object belongs to (outputting 1 for target class and 0 for background class). Provided we precisely knew the distribution of measurements for both classes and prior probability of the object being a target it can be proved that the algorithm which minimizes the probability of classification error is the likelihood ratio detector.

To be more formal, let us have the finite alphabet $\{0, 1, \ldots, F\}$ and a collection of measurements $(X_1, \ldots, X_L)$. Since we assume them to be independent it will be more practical to work with a histogram $\vec{x}$ (treating it as $F+1$-dimensional vector) defined as:

$$x_k = \#\{(X_l, l) \mid X_l = k,\ 1 \le l \le L\}, \qquad 0 \le k \le F \quad (1)$$

This paper was supported by a subaward agreement from The Johns Hopkins University with funds provided by Grant No. IIS-0121285 from The National Science Foundation. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of JHU or The National Science Foundation.

This work was also supported by the project LN00A063 of the Ministry of Education of the Czech Republic.

Note that $\mathrm{L}(\vec{x}) := \sum_0^F x_k = L$. Let $T = (T_0, \ldots, T_F)$ be the probability distribution of measurements for targets and $B = (B_0, \ldots, B_F)$ the same for backgrounds. Let α be the prior probability that the observed object is from target class and $\vec{x}$ be the histogram of measurements $(X_1, \ldots, X_L)$ on the observed object. Then the likelihood ratio detector is given by

$$c((x_0, \ldots, x_F), \alpha, T, B) := 1 \iff \frac{\Pr(X_1, \ldots, X_L|T)}{\Pr(X_1, \ldots, X_L|B)} = \prod_{k=0}^{F} \frac{{T_k}^{x_k}}{{B_k}^{x_k}} \ge \frac{(1-\alpha)}{\alpha} \quad (2)$$

and it minimizes the probability of classification error.

2. The new optimal detector

The main drawback of the preceding detector (apart from the independence assumption) is the requirement of the precise knowledge of the distributions. Although they can be estimated from the training data, the performance tends to be poor if we have a few of them.

The detector described in this section will treat this problem. Instead of being a function of the input utterance and the respective distributions $c(x, \alpha, T, B)$ it will be a function of the input utterance and the training data. Formally, it will be a (deterministic) function $c(\vec{x}, \alpha, \vec{t}, \vec{b})$, where $\vec{t}$ and $\vec{b}$ is the training data for target and background respectively.

Let us characterize the optimality in this case. We will assume that both the target and the background have some underlying distributions T and B which are unknown to us. The only thing we actually know are the training data $\vec{t}$ and $\vec{b}$. Since we assume the independence, we can consider $\vec{t}$ to be a histogram of a random draw from T. Similarly we consider $\vec{b}$ to be a histogram of a random draw from B.

2.1 Observation *For the fixed* $\mathrm{L}(\vec{t})$*, the probability of* $\vec{t}$ *to be drawn from* T *is*[1]

$$\Pr(\ \vec{t}\ \text{drawn from}\ T \mid \mathrm{L}(\vec{t}) = L_t\ ) = \begin{bmatrix} L_t \\ t_0, \ldots, t_F \end{bmatrix} \cdot \prod_{k=0}^{F} T_k^{t_k} \quad (3)$$

where we put $0^0 := 1$ *for sake of brevity.*

The optimality criterion will be based on minimizing the probability of detection error. But we have to be more careful here since for a fixed T and B the optimal detector is the one using the likelihood ratio. Thus the new detector would

[1] the symbol $\left[\begin{smallmatrix} a_1, \ldots, a_n \\ b_1, \ldots, b_m \end{smallmatrix}\right]$ is defined as $\frac{a_1! \cdots a_n!}{b_1! \cdots b_m!}$; note that $\binom{n}{k} = \left[\begin{smallmatrix} n \\ k, n-k \end{smallmatrix}\right]$ for $n \ge k \ge 0$; n,k natural numbers

be suboptimal were we using literally the same criterion. The key point is that since we *don't know* the true distributions, we will optimize for the average performance. To be more specific let's imagine how the detector's performance could be tested. There could be the examiner who would simulate the real use of the detector by giving it randomly selected training and testing data and by counting the detector's errors thus estimating its probability of the detection error. To formalize this, we will introduce the examining procedure, which lends itself for deriving the probability of error analytically, what will enable us to design the detector such that this probability of error would be minimized.

Examining Procedure: There is a prior probability α known both to the examiner and to the detector. In the first step, the examiner randomly chooses the distributions T and B. Since we want the detector to be completely data-driven we assume that the examiner chooses the distribution uniformly (from the set of all distributions over the given finite alphabet). This could be generalized later by allowing some distributions to be more probable than others thus incorporating some prior knowledge (such as the Zipf's law) into the detector. But let's keep things simple now. In the second step the examiner randomly draws training streams from T and B. We will write their histograms as $\vec{t}$ and $\vec{b}$. Then, with probability α he draws testing data with the histogram $\vec{x}$ form T, otherwise (with probability $1-\alpha$) from B. The probability of error for the fixed α can be written as follows:

$$\Pr\left(\text{error}\right) = \sum_{\vec{x},\vec{t},\vec{b}} \Pr\left(\text{error on } \vec{x},\vec{t},\vec{b}\,\right) \tag{4}$$

where we will minimize

$$\begin{aligned}\Pr\left(\text{error on } \vec{x},\vec{t},\vec{b}\,\right) = c(\vec{x},\alpha,\vec{t},\vec{b}\,)\cdot(1-\alpha)\cdot \qquad (5)\\ \cdot \Pr\left(\vec{x} \text{ and } \vec{b} \text{ drawn from } B,\, \vec{t} \text{ from } T;\, T,\, B \text{ unknown}\right)\\ + \quad \big(1-c(\vec{x},\alpha,\vec{t},\vec{b}\,)\big)\cdot\alpha\cdot\\ \cdot \Pr\left(\vec{b} \text{ drawn from } B,\, \vec{x} \text{ and } \vec{t} \text{ from } T;\, T,\, B \text{ unknown}\right)\end{aligned}$$

for every single $(\vec{x},\vec{t},\vec{b}\,)$ separately[2]. Essentially this number is the probability that the examiner came up with some data $(\vec{x},\vec{t},\vec{b}\,)$ and the detector claimed $\vec{x}$ to be from the different class than the examiner obtained it from (which is what we call a classification error). Since for a fixed $(\vec{x},\vec{t},\vec{b}\,)$ we have

$$\begin{aligned}&\Pr\left(\vec{x} \text{ and } \vec{b} \text{ drawn from } B,\, \vec{t} \text{ drawn from } T;\, T,\, B \text{ unknown}\right) =\\ &= \sum_{L_x,L_t,L_b} \Pr\left(\vec{x} \text{ and } \vec{b} \text{ drawn from } B,\, \vec{t} \text{ drawn from } T;\right. \qquad (6)\\ &\left.;T,\, B \text{ unknown and } \underbrace{\mathrm{L}(\vec{x})=L_x,\, \mathrm{L}(\vec{t})=L_t,\, \mathrm{L}(\vec{b})=L_b}_{\bullet}\right)\\ &= \sum_{L_x,L_t,L_b} \Pr\left(\vec{x},\, \vec{b} \text{ from } B,\, \vec{t} \text{ from } T;\, T,\, B \text{ unknown}|\bullet\right)\cdot\Pr\left(\bullet\right)\end{aligned}$$

we can write

$$\Pr\left(\text{error on } \vec{x},\vec{t},\vec{b}\,\right) = \sum_{L_x,L_t,L_b} P_e(\vec{x},\vec{t},\vec{b}\,)\cdot\Pr\left(\bullet\right) \tag{7}$$

[2]first, this will bring us to the minimum of the original formula since it is a sum of numbers and we are minimizing each of them separately; second, we can do it, since for each term we have unique $(\vec{x},\vec{t},\vec{b}\,)$ "identifying" it, and $c(\cdot)$ depends right on this identification — so we can always choose c to be 0 or 1 to minimize the term independently of other terms since they have different "identification"

where

$$\begin{aligned}P_e(\vec{x},\vec{t},\vec{b}\,) = c(\vec{x},\alpha,\vec{t},\vec{b}\,)\cdot(1-\alpha)\cdot F(\vec{x},\vec{t},\vec{b}\,) \qquad (8)\\ +\big(1-c(\vec{x},\alpha,\vec{t},\vec{b}\,)\big)\cdot\alpha\cdot M(\vec{x},\vec{t},\vec{b}\,)\end{aligned}$$

where

$$\begin{aligned}F(\vec{x},\vec{t},\vec{b}\,) = \Pr\left(\vec{x},\vec{b} \text{ from } B,\, \vec{t} \text{ from } T;\, T,B \text{ unknown}|\bullet\right) \qquad (9)\\ M(\vec{x},\vec{t},\vec{b}\,) = \Pr\left(\vec{b} \text{ from } B,\, \vec{x},\vec{t} \text{ from } T;\, T,B \text{ unknown}|\bullet\right)\end{aligned}$$

The term $\Pr\left(\bullet\right)$ is unknown and depending solely on the examiner's will. From the point of view of the detector its distribution is fixed, so we can get to the minimum by minimizing $P_e(\cdot)$ only. We can leave the constant term $\alpha M(\vec{x},\vec{t},\vec{b}\,)$ out, since the minimum does not depend on it, and obtain

$$\begin{aligned}&P'_e(\vec{x},\vec{t},\vec{b}\,) := \qquad (10)\\ &c(\vec{x},\alpha,\vec{t},\vec{b}\,)\Big((1-\alpha)F(\vec{x},\vec{t},\vec{b}\,)-\alpha M(\vec{x},\vec{t},\vec{b}\,)\Big)\end{aligned}$$

The right parenthesis takes positive and negative values, while the detector's output is either 0 or 1. Therefore, to make the formula minimal it suffices to "use" all the negative parenthesis and zero-out the positive ones. This can be done since each parenthesis is determined by detector's argument. The optimal detector is given by the following formula:

$$c(\vec{x},\alpha,\vec{t},\vec{b}\,) := 1 \iff \frac{M(\vec{x},\vec{t},\vec{b}\,)}{F(\vec{x},\vec{t},\vec{b}\,)} \geq \frac{1-\alpha}{\alpha} \tag{11}$$

In the rest of this section we will be computing probabilities $M(\cdot)$ and $F(\cdot)$ to obtain the final formula for the detector. Note that

$$\begin{aligned}F(\vec{x},\vec{t},\vec{b}\,) := \Pr\left(\vec{t} \text{ drawn from } T,\, T \text{ unknown} \mid\right. \qquad (12)\\ \left.\mathrm{L}(\vec{t})=L_t\right)\cdot\Pr\left(\vec{x} \text{ and } \vec{b} \text{ from } B,\, B \text{ unknown} \mid\right.\\ \left.\mathrm{L}(\vec{x})=L_x,\, \mathrm{L}(\vec{b})=L_b\right)\end{aligned}$$

because of independence.

2.2 Definition The set $\mathrm{Splx}_F = \{(T_1,\ldots,T_F) \mid 0<T_k<1,\, \sum_1^F T_k<1\}$ will be called the simplex.

The simplex is a set of all possible distributions $(D_0,\ldots,D_F)$ on $F+1$ features, provided we define $D_0 = 1-\sum_{k=1}^F D_k$ and take $(D_1,\ldots,D_F)\in\mathrm{Splx}_F$. Therefore the probability

$$\Pr\left(\vec{t} \text{ drawn from } T,\, T \text{ unknown} \mid \mathrm{L}(\vec{t})=L_t\right) \tag{13}$$

can be expressed as

$$\int\limits_{T\in\mathrm{Splx}_F} \Pr\left(\vec{t} \text{ drawn from } T \mid \mathrm{L}(\vec{t})=L_t\right)\rho(T)dT \tag{14}$$

where $\rho(T)$ is probability density function of the distribution T. As we already noted we define $\rho(T)$ to be a constant C_F. Using observation 2.1 we get

$$\begin{aligned}&\int\limits_{T\in\mathrm{Splx}_F} \Pr\left(\vec{t} \text{ drawn from } T \mid \mathrm{L}(\vec{t})=L_t\right)\cdot C_F\cdot dT = \qquad (15)\\ &C_F\cdot\int\limits_{\substack{(T_1,\ldots,T_F)\in\mathrm{Splx}_F\\ T_0=1-\sum_1^F T_k}} \begin{bmatrix}\mathrm{L}(\vec{t})\\ t_0,\ldots,t_F\end{bmatrix}\cdot\prod_{k=0}^F T_k^{t_k}\, d(T_1,\ldots,T_F)\\ &= C_F\cdot\begin{bmatrix}\mathrm{L}(\vec{t})\\ t_0,\ldots,t_F\end{bmatrix} I(t_0,\ldots,t_F)\end{aligned}$$

where

$$I(t_0,\ldots,t_F) := \int\limits_{\substack{(T_1,\ldots,T_F)\in \mathrm{Splx}_F \\ T_0=1-\sum_1^F T_k}} \prod_{k=0}^{F} T_k^{t_k}\, d(T_1,\ldots,T_F) \quad (16)$$

Similarly we get:

$$\Pr\left(\vec{x} \text{ and } \vec{b} \text{ from } B,\ B \text{ unknown} \mid \mathrm{L}(\vec{x}) = L_x,\ \mathrm{L}(\vec{b}) = L_b\right) = C_F \cdot \int\limits_{\substack{(B_1,\ldots,B_F)\in \mathrm{Splx}_F \\ B_0=1-\sum_1^F B_k}} \begin{bmatrix} \mathrm{L}(\vec{b}) \\ b_0,\ldots,b_F \end{bmatrix} \begin{bmatrix} \mathrm{L}(\vec{x}) \\ x_0,\ldots,x_F \end{bmatrix} \prod_{k=0}^{F} B_k^{x_k+b_k} \cdot d(B_1,\ldots,B_F) \quad (17)$$

because of independence of drawing $\vec{x}$ and $\vec{b}$ from B. This leads to

$$C_F \begin{bmatrix} \mathrm{L}(\vec{b}) \\ b_0,\ldots,b_F \end{bmatrix} \begin{bmatrix} \mathrm{L}(\vec{x}) \\ x_0,\ldots,x_F \end{bmatrix} I(x_0+b_0,\ldots,x_F+b_F) \quad (18)$$

So it turns out that all we have to do is to compute the integral $I(\cdot)$.

2.3 Lemma

$$\int\limits_{\substack{(T_1,\ldots,T_F)\in \mathrm{Splx}_F \\ T_0=1-\sum_1^F T_k}} \prod_{k=0}^{F} T_k^{t_k}\, d(T_1,\ldots,T_F) = \begin{bmatrix} t_0,\ldots,t_F \\ \mathrm{L}(\vec{t}) + F \end{bmatrix} \quad (19)$$

Proof Using a suitable change of variables we can transform this integral to a product of integrals of the same kind with $F = 1$ which is essentially the definition for Euler's beta function. See [2] for details. Q.E.D.

Now, substituting

$$\Pr\left(\vec{t} \text{ drawn from } T,\ T \text{ unknown} \mid \mathrm{L}(\vec{t}) = L_t\right) = \quad (20)$$

$$C_F \cdot \begin{bmatrix} \mathrm{L}(\vec{t}) \\ t_0,\ldots,t_F \end{bmatrix} \cdot \begin{bmatrix} t_0,\ldots,t_F \\ \mathrm{L}(\vec{t}) + F \end{bmatrix} = C_F \begin{bmatrix} \mathrm{L}(\vec{t}) \\ \mathrm{L}(\vec{t}) + F \end{bmatrix}$$

$$\Pr\left(\vec{x} \text{ and } \vec{b} \text{ from } B,\ B \text{ unknown} \mid \mathrm{L}(\vec{x}) = L_x,\ \mathrm{L}(\vec{b}) = L_b\right) = C_F \begin{bmatrix} \mathrm{L}(\vec{b}) \\ b_0,\ldots,b_F \end{bmatrix} \begin{bmatrix} \mathrm{L}(\vec{x}) \\ x_0,\ldots,x_F \end{bmatrix} \begin{bmatrix} x_0+b_0,\ldots,x_F+b_F \\ \mathrm{L}(\vec{x}) + \mathrm{L}(\vec{b}) + F \end{bmatrix} = C_F \begin{bmatrix} \mathrm{L}(\vec{b}),\ \mathrm{L}(\vec{x}),\ x_0+b_0,\ldots,x_F+b_F \\ \mathrm{L}(\vec{x}) + \mathrm{L}(\vec{b}) + F,\ x_0,\ldots,x_F,\ b_0,\ldots,b_F \end{bmatrix}$$

into the definition of $F(\vec{x},\vec{t},\vec{b}\,)$ we end up with

$$F(\vec{x},\vec{t},\vec{b}\,) = {C_F}^2 \cdot \begin{bmatrix} \mathrm{L}(\vec{t}) \\ \mathrm{L}(\vec{t}) + F \end{bmatrix} \cdot \quad (21)$$

$$\cdot \begin{bmatrix} \mathrm{L}(\vec{b}),\ \mathrm{L}(\vec{x}),\ x_0+b_0,\ldots,x_F+b_F \\ \mathrm{L}(\vec{x}) + \mathrm{L}(\vec{b}) + F,\ x_0,\ldots,x_F,\ b_0,\ldots,b_F \end{bmatrix}$$

Similarly we could derive that

$$M(\vec{x},\vec{t},\vec{b}\,) = {C_F}^2 \cdot \begin{bmatrix} \mathrm{L}(\vec{b}) \\ \mathrm{L}(\vec{b}) + F \end{bmatrix} \cdot \quad (22)$$

$$\cdot \begin{bmatrix} \mathrm{L}(\vec{t}),\ \mathrm{L}(\vec{x}),\ x_0+t_0,\ldots,x_F+t_F \\ \mathrm{L}(\vec{x}) + \mathrm{L}(\vec{t}) + F,\ x_0,\ldots,x_F,\ t_0,\ldots,t_F \end{bmatrix}$$

Hence

$$\mathrm{Score}(\vec{x},\vec{t},\vec{b}\,) := \frac{M(\vec{x},\vec{t},\vec{b}\,)}{F(\vec{x},\vec{t},\vec{b}\,)} = \quad (23)$$

$$= \frac{\begin{bmatrix} x_0+t_0,\ldots,x_F+t_F \\ t_0,\ldots,t_F \end{bmatrix}}{\begin{bmatrix} x_0+b_0,\ldots,x_F+b_F \\ b_0,\ldots,b_F \end{bmatrix}} \cdot \frac{\begin{bmatrix} \mathrm{L}(\vec{t}) + F \\ \mathrm{L}(\vec{x}) + \mathrm{L}(\vec{t}) + F \end{bmatrix}}{\begin{bmatrix} \mathrm{L}(\vec{b}) + F \\ \mathrm{L}(\vec{x}) + \mathrm{L}(\vec{b}) + F \end{bmatrix}}$$

The final formula defining the detector is then

$$c(\vec{x},\alpha,\vec{t},\vec{b}\,) := 1 \quad \Longleftrightarrow \quad \mathrm{Score}(\vec{x},\vec{t},\vec{b}\,) \geq \frac{1-\alpha}{\alpha} \quad (24)$$

and it is optimal in the above sense.

3. Asymptotic behavior

3.1 Theorem *The more training data do we have, the more the new detector approximates the old one, provided that T and B distributions have non-zero probabilities for all features. Formally, for a fixed $\vec{x}$ and α:*

$$\forall \varepsilon > 0 \lim_{(\mathrm{L}(\vec{t}),\mathrm{L}(\vec{b}))\to(\infty,\infty)} \quad (25)$$

$$\Pr\left(\left|\log\big(\mathrm{Score}(\vec{x},\vec{t},\vec{b}\,)\big) - \sum_{k=0}^{F} x_k \cdot \log\left(\frac{T_k}{B_k}\right)\right| < \varepsilon\right) = 1$$

where $\vec{t}$ is drawn form T, $\vec{b}$ is drawn from B and $B_k \cdot T_k > 0$ for all k. The term with the sum sign is the log version of the likelihood ratio detector.

Proof See [2]. Q.E.D.

3.2 Corollary *The probability of error of the new detector converges to the probability of error of the old one as we have more training data (assuming $T_k \cdot B_k > 0$ for each k). Formally, for a fixed $\vec{x}$, T, B and α:*

$$\lim_{N\to\infty} \Pr\left(\text{new detector makes an error on } (\vec{x},\alpha,\vec{t},\vec{b}),\right. \quad (26)$$

$$\left.\vec{t} \text{ from } T,\ \vec{b} \text{ form } B,\ \mathrm{L}(\vec{b}) > N \text{ and } \mathrm{L}(\vec{t}) > N\ \right) =$$

$$\Pr\left(\text{old detector makes an error on } (\vec{x},\alpha,T,B)\right)$$

Proof See [2]. Q.E.D.

4. Practical tests

We tested this detector on the problem of speaker detection. We conducted the experiments on Switchboard-1 corpus according to the NIST Extended Data paradigm. In this paradigm there are 5 training conditions consisting of 1, 2, 4, 8 and 16 conversation sides, where a side is nominally 2.5 minutes in duration. Testing is done on entire conversation side. The evaluation consists of a 6-split jack-knife over the entire corpus. Two background models were used for testing. One model was trained using data from splits 1-3 and applied when testing on splits 4-6, the other was trained on splits 4-6 and used on splits 1-3. We used the open-loop phone recognizer Lincoln PPRLM-LID which provided us with the English phone stream representation of the given utterance. We further removed silences detected by Lincoln-Lab speech activity detector before using this

stream. Thus a single measurement from the detector's point of view was one of 47 possible phones appearing in a given 1/100 second frame, while the collection of measurements was a histogram of single measurements taken over all non-silence and non-noise frames of the utterance.

We first determined the Equal Error Rates of the original likelihood ratio detector as our baseline:

Table 1: *EERs of the original (likelihood ratio) detector*

1	2	4	8	16
15.17%	12.22%	9.99%	9.14%	9.22%

Each column represents given training condition as denoted in its heading by a number of training conversations. It can be seen that the performance degrades as we have less training data.

Then we ran the new detector on the same data. In the implementation we actually computed log-score by summing the log-factorials up to number 1000. For higher numbers the Stirling approximation was used. The results were surprising, unfortunately:

Table 2: *EERs of the new detector*

1	2	4	8	16
27.49%	20.12%	14.46%	11.17%	10.10%

Nearly two times as worse at the 1 conversation training than the old detector. Ironically the new detector was designed to help especially this case. It was found that the reason for this behavior is in the assumption of mutual independence of the measurements. Although the likelihood ratio detector also depends on this assumption it is seemingly more resistant to its violation than the new detector. On the other hand new detector is able to outperform the old one when we make measurements "less dependent". This can be done by a random draw of the frames from the whole utterance. We did an experiment using the new detector, where we draw some random frames from the whole conversation using them as the input for the detector. We found out that the performance drop is quite modest compared to how small portion of the file we actually used. In each utterance evaluation we took randomly 96 frames from the tested utterance, 48 frames from target training data and 48 frames from background training data. Totally less than 2 seconds of non-silence sound scattered (pseudo) randomly over the original data:

Table 3: *EERs of the new detector; random draw*

1	2	4	8	16
39.08%	39.88%	38.36%	39.52%	40.22%

Table 4: *EERs of the old detector; random draw*

1	2	4	8	16
44.87%	45.55%	45.16%	44.51%	46.82%

That means that the new detector achieved $39.06/27.49 = 1.42$ times worse EER, while the old one is $44.87/15.17 = 2.96$ times worse. In this particular case the new detector outperformed the old one $44.87/39.06 = 1.15$ times.

When we took 10 random samples of this size and averaged the log-scores, the results were even better:

Table 5: *EERs of the new detector; 10 random draws*

1	2	4	8	16
26.45%	25.56%	24.26%	24.49%	25.04%

Table 6: *EERs of the old detector; 10 random draws*

1	2	4	8	16
38.31%	37.94%	37.19%	36.78%	37.52%

Here, the new detector outperformed the old one by the factor of $38.31/26.45 = 1.45$.

5. Fighting the statistical dependence

The results of practical tests with small samples indicate that we could take many small scattered (hence nearly independent) samples out of the utterance and compute the optimal scores on all of them. Then we could use the average score value for the detector's decision. For example if we knew that the features 80 frames apart are nearly independent we could take random samples of size $\mathrm{L}(\vec{x})/160$ satisfying the distance constrain. The drawback is that this is computationally intractable since we would need dozens of trials. But we could try to solve this analytically. To ease the problem let us sum *all* of the possible samples of the given size (this allows us to completely forget the spatial arrangement of the frames in the utterance and work with histograms as before). For simplicity of the formulas, we will further allow the same frame to be selected multiple times into a given sample. Provided the sample size is reasonably small in comparison with the utterance length, there will be majority of the samples satisfying the distance constrain. The rest of them will not disturb the score value too much then.

We'll introduce the averaging formula for the new optimal detector in this section. Let us have histograms of not necessarily independent training data $\vec{t}$ and $\vec{b}$ and the histogram of the tested utterance $\vec{x}$. Let us choose the numbers specifying the size of the samples by $L_x \le \mathrm{L}(\vec{x})$, $L_t \le \mathrm{L}(\vec{t})$ and $L_b \le \mathrm{L}(\vec{b})$. Denote $y_k = \hat{x}_k / \mathrm{L}(\vec{x})$, $v_k = \hat{t}_k / \mathrm{L}(\vec{t})$ and $w_k = \hat{b}_k / \mathrm{L}(\vec{b})$. Then the averaged log-score computed over all samples of the length L_x, L_t and L_b respectively is given by the following formula:

$$\begin{aligned}\mathrm{ALscore}(L_x, L_t, L_b, \vec{y}, \vec{v}, \vec{w}) := \sum_{\substack{\vec{x},\vec{t},\vec{b} \\ \Sigma x_k = L_x \\ \Sigma t_k = L_t \\ \Sigma b_k = L_b}} \begin{bmatrix} L_x \\ x_0, \ldots, x_F \end{bmatrix} \cdot \\ \cdot \left(\prod_{k=0}^{F} {y_k}^{x_k} \right) \cdot \begin{bmatrix} L_t \\ t_0, \ldots, t_F \end{bmatrix} \cdot \left(\prod_{k=0}^{F} {v_k}^{t_k} \right) \cdot \\ \cdot \begin{bmatrix} L_b \\ b_0, \ldots, b_F \end{bmatrix} \cdot \left(\prod_{k=0}^{F} {w_k}^{b_k} \right) \cdot \\ \cdot \log \left(\frac{\begin{bmatrix} x_0 + t_0, \ldots, x_F + t_F \\ t_0, \ldots, t_F \end{bmatrix}}{\begin{bmatrix} x_0 + b_0, \ldots, x_F + b_F \\ b_0, \ldots, b_F \end{bmatrix}} \cdot \frac{\begin{bmatrix} L_t + F \\ L_x + L_t + F \end{bmatrix}}{\begin{bmatrix} L_b + F \\ L_x + L_b + F \end{bmatrix}} \right)\end{aligned} \tag{27}$$

Due to the multinomial expansion we can write

$$\begin{aligned}&\text{ALscore}(L_x, L_t, L_b, \vec{y}, \vec{v}, \vec{w}) = \\ &\text{HalfALscore}(L_x, L_t, \vec{y}, \vec{v}) - \text{HalfALscore}(L_x, L_b, \vec{y}, \vec{w})\end{aligned} \tag{28}$$

where

$$\begin{aligned}\text{HalfALscore}(L_x, L_t, \vec{y}, \vec{v}) := \sum_{\substack{\vec{x},\vec{t} \\ \Sigma x_k = L_x \\ \Sigma t_k = L_t}} &\begin{bmatrix} L_x \\ x_0, \ldots, x_F \end{bmatrix} \cdot \\ \cdot \begin{bmatrix} L_t \\ t_0, \ldots, t_F \end{bmatrix} &\cdot \left(\prod_{k=0}^{F} {y_k}^{x_k} \cdot {v_k}^{t_k} \right) \cdot \\ \cdot \log \left(\begin{bmatrix} x_0 + t_0, \ldots, x_F + t_F \\ t_0, \ldots, t_F \end{bmatrix} \right. &\left. \cdot \begin{bmatrix} L_t + F \\ L_x + L_t + F \end{bmatrix} \right)\end{aligned} \tag{29}$$

After rather technical manipulations (expanding the products, using the discrete Fourier transform and convolution lemma) we get the final formula:

$$\begin{aligned}&\text{HalfALscore}(L_x, L_t, \vec{y}, \vec{v}) = \\ &\log \begin{bmatrix} L_t + F \\ L_x + L_t + F \end{bmatrix} + \sum_{p=0}^{F} D(p)\end{aligned} \tag{30}$$

where

$$\begin{aligned}D(p) := &-c_0 \cdot (1 - v_p)^{L_t} \left((1 - y_p)^{L_x} - 1 \right) + \\ &\sum_{f=1}^{N/2-1} 2\, C_f \,\text{Re}\Big(\left(1 - v_p + v_p e^{2\pi i f/N} \right)^{L_t} \cdot \\ &\cdot \Big(\left(1 - y_p + y_p e^{2\pi i f/N} \right)^{L_x} - 1 \Big) \Big)\end{aligned} \tag{31}$$

and where N is a number divisible by 4, $N \geq 2(L_x + L_t + 1)$, C_f is the discrete Fourier transform of c_k defined as follows:

$$\begin{aligned}c_k &:= c_{N-k} := \log(k!) \qquad \text{for} \quad 0 < k < \frac{N}{2} \\ c_{\frac{N}{2}} &:= \frac{1}{2} \Big(\log(\frac{N}{2}!) + \log((\frac{N}{2} - 1)!) \Big) \\ c_0 &:= -\sum_{k=1}^{N-1} (-1)^k c_k\end{aligned} \tag{32}$$

Note that C_f can be precomputed during initialization so it is a matter of table look-up in the runtime. In our implementation we used several tables, each for a distinct $N = 2^r$ (precomputed using FFT). This caused that the detector had to add at most 2 times longer sequence than $L_x + L_t$ was. Since C_f falls to zero as f goes to $N/2$, we compute the sum in the backward direction to reduce roundoff error. Note that $D(p) = 0$ whenever $y_p = 0$; such terms can be omitted altogether. Note that the corresponding formula for HalfALscore$(L_x, L_b, \vec{y}, \vec{v})$ can be obtained from formula 30 by substituting L_b for L_t.

6. Averaging likelihood ratio detector

We will not go into the details here and only remark that this averaging can be done (more easily) to the old detector too. As a surprise, the result of it is the old detector itself just with the score value divided by the length of the tested data (L$(\vec{x})$). First, it means that the result of averaging does not depend on the sizes of random draw (L_x, L_t and L_b). Second, it means that in case where all the tested data have approximately the same length, the original likelihood ratio detector exhibits less sensitivity to statistical dependence (because if the lengths are the same, the original detector makes the same decisions as the averaged one, provided that α parameter is scaled properly). This may explain the old detector's resistance to the independence assumption violation as observed in case of speaker detection.

7. Performance of the averaged new detector

The final score value depends on the sizes of samples we are drawing from the histograms. Unfortunately we don't have sufficiently developed theory which could tell us the right values of sample sizes L_x, L_t and L_b based on the length of currently tested utterance and the sizes of the training data, or at least which could give us some constraints on their values. That is why we did a simple repeated line search on split 3 (the one being the hardest for this detector) repeatedly evaluating it for different values of L_x while the other sizes (L_b, L_t) were fixed. After we found a minimum, we fixed L_x here and let L_t change instead. Finally a minimum for changing L_b was found. Note that sample sizes (L_x, L_b, L_t) were fixed here throughout each evaluation of the split, not depending on the length of the utterance and the size of the training data. We did two rounds of this procedure and discovered local optimum at $L_x = 300$, $L_t = 250$ and $L_b = 145$ with the following performance (now evaluated using all splits):

Table 7: *EERs of the new averaged detector*

1	2	4	8	16
14.03%	11.05%	9.07%	7.78%	8.27%

While the old detector averaged as described in the previous section achieved:

Table 8: *EERs of the old averaged detector*

1	2	4	8	16
15.05%	12.17%	9.88%	9.06%	9.46%

Note that the detector outperformed the old averaged detector by a factor of 15.05/14.03=1.07 at 1 conversation training and rather surprisingly by 9.06/7.78=1.16 at 8 conversation training. However, we found out that the area of the minimum is quite narrow and, when missed, the performance falls far behind the old detector. The DET plots of the averaged new detector are in the figure 1.

8. Discussion

Let us note first, that the averaged detector is no longer optimal. As we saw above it is able to outperform the old detector when the parameters are properly tuned. The tuning itself requires further research, however. Also other questions arise, for example what is the proper score-fusion method for this kind of detectors.

For these reasons the detector was not used as a replacement for the likelihood ratio detector in the real-world detection problems yet. The intended use for it was in the conditional pronunciation modeling speaker detection method (as introduced in [1]).

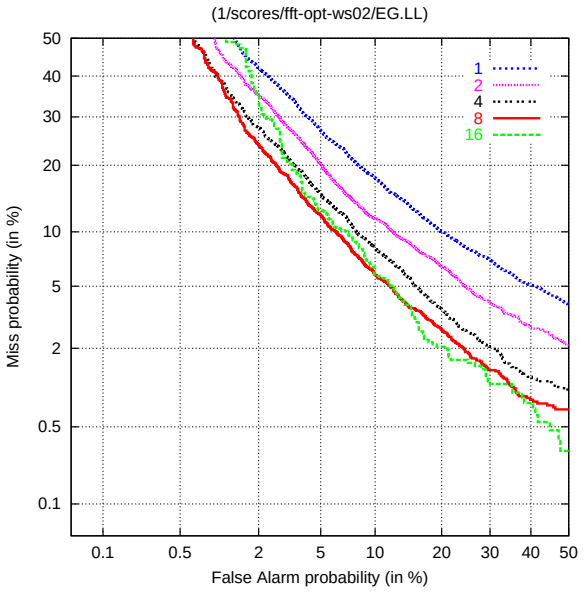

Figure 1: *The DET curves of the averaged optimal detector.*

We still have a few pages left, so we will briefly describe this method here, because its EERs are now almost two times better than it is described in [1]. The principle of the method remains unchanged, however. Better feature selection is responsible for the improvement.

9. Principle of the pronunciation modeling

Conditional pronunciation modeling (CPM) was developed at JHU/CLSP Summer Workshop 2002 and it is a method for open-set speaker detection task that does not rely on acoustic vectors. Instead it tries to use higher-level information carried by the speech signal. By a "pronunciation model" is meant an ambiguous "mapping" from phonemes that the speaker intends to pronounce to phones he actually pronounced. Since we don't know speaker's intention we use ASR to decode words spoken followed by lookup of phonemes in the lexicon (one word can have multiple lexicon pronunciations). Then these lexicon pronunciations are force-aligned with audio and the one with the highest score is selected as the phonetic transcription of the word. This is how time aligned phoneme stream is computed. Phones actually pronounced are taken from open-loop phone recognizer. These open-loop phones alone were used in the previous testing of the averaged new detector. Pronunciation modeling estimates conditional probabilities of open-loop phones given the ASR-phonemes (that is "reality" given the "intention") on per frame basis. Then, standard likelihood ratio detector is applied to these probabilities, as will be described in the next two sections.

The relation of what has been said (phonemes) versus how it has been said (phones) is crucial to the pronunciation modeling method; see [1] for details.

10. Training

Since we are using likelihood ratio detector we have two models: the background model (P_{BG}) and the speaker model (P_{SP}). Both consist of conditional probability estimates of open-loop phones given the ASR phonemes and are trained in the same way:

$$P(e \mid a) := \frac{\#\{(e,a,t) \mid \exists\, t : (e,a,t) \in \text{INPUT}\}}{\#\{(x,a,t) \mid \exists\, x,t : (x,a,t) \in \text{INPUT}\}} \quad (33)$$

Where INPUT is a set of (E, A, F) triples where the first element is the open-loop phone recognized in frame F, while the second is the force aligned phoneme assigned to frame F. We further assume that INPUT (both in training and testing) has been filtered by removing frames which ASR marked as a silence and frames containing crosstalk. No thresholding nor smoothing of the estimated probabilities is being used in the current implementation.

11. Testing

Score of the test utterance is computed as follows:

$$\text{Score} = \frac{1}{\#M} \sum_{(e,a,t)\in M} (\log(P_{\mathrm{SP}}(e|a)) - \log(P_{\mathrm{BG}}(e|a))) \quad (34)$$

where

$$M = \{(e,a,t) \mid (e,a,t) \in \text{INPUT s.t. both } P_{\mathrm{SP}}(e|a) \text{ and } P_{\mathrm{BG}}(e|a) \text{ are defined}\} \quad (35)$$

This means that only those pairs (e, a) that has been seen during the training of both speaker and background model are counted. The slight difference from the original method is in the presence of factor $\frac{1}{\#M}$ which lowers the effects of statistical dependence among frames.

12. Split phones

At WS02 (see [4, 5]) we devised an ad-hoc but working enhancement. Instead of using whole phonemes to condition to, we would like to condition to the HMM states of ASR recognizer. Unfortunately, these were not available to us, so we did an ad-hoc labeling of the respective frames of the phoneme by four labels as *short/head/body/tail*. Thus we obtained 4 times larger phone alphabet and we could use the same algorithm on the respective preprocessed data. See [1] for more detailed specification.

13. Experiments

Experiments were conducted on the Switchboard-I corpus according to the NIST Extended Data paradigm. We used phone sequences from five languages: English (EG), German (GE), Spanish (SP), Japanese (JA), and Mandarin (MA). The phone recognizer is from the Lincoln PPRLM LID system and uses gender-dependent phone models. Its output is a time-aligned phone stream. For phoneme stream we used time aligned phoneme output from SRI ASR system (30% WER).

We re-implemented the method in the C language (originally we used Perl at WS02). We also abandoned the text coding of the phone files. The new binary coding made the representation of the SWB database roughly 50 times smaller allowing it to fit into the OS disk cache. These two factors are responsible for a significant speed-up. The whole SWB-1 training and testing (all splits, all training conversation conditions) took about several hours at WS02 using a cluster of 5 computers. Now it takes about 20 to 40 seconds on a single 1.5 GHz machine. This speedup allowed us to use "brute force" to test

Table 9: *EERs of best CPM systems, part 1*

1	2
9.78%-J-JA.ASR	5.06%-J-GE.ASR.LL
9.78%-J-JA.ASR.ASR	5.25%-J-GE.ASR
9.80%-J-JA.ASR.LL	5.25%-J-GE.ASR.ASR
9.84%-J-SP.ASR.LL	5.38%-J-EG.ASR.LL
10.07%-J-SP.ASR	5.50%-J-JA.sASR.LL
10.07%-J-SP.ASR.ASR	5.55%-J-EG.ASR
10.80%-J-MA.ASR.LL	5.55%-J-EG.ASR.ASR
10.82%-J-MA.ASR	5.59%-J-JA.ASR.LL
10.82%-J-MA.ASR.ASR	5.63%-J-MA.ASR.LL
10.99%-J-GE.ASR.LL	5.65%-J-MA.ASR
11.07%-J-GE.ASR	5.65%-J-MA.ASR.ASR
11.07%-J-GE.ASR.ASR	5.69%-J-SP.ASR.LL
11.09%-J-EG.JA.ASR	5.76%-ASR.MA
11.28%-J-EG.SP.ASR	5.76%-ASR.MA.ASR
11.28%-JA.ASR.LL	5.78%-J-JA.sASR
11.32%-JA.ASR	5.78%-J-JA.sASR.ASR
11.32%-JA.ASR.ASR	5.80%-J-SP.ASR
11.36%-J-JA.GE.ASR	5.80%-J-SP.ASR.ASR

Table 10: *EERs of best CPM systems, part 2*

4	8
2.47%-sASR.JA.LL	1.44%-sASR.MA
2.54%-sASR.JA	1.44%-sASR.MA.ASR
2.54%-sASR.JA.ASR	1.47%-sASR.GE
2.54%-sASR.MA.LL	1.47%-sASR.GE.ASR
2.60%-J-MA.sASR.LL	1.52%-sASR.MA.LL
2.60%-sASR.MA	1.57%-sASR.GE.LL
2.60%-sASR.MA.ASR	1.65%-sASR.JA
2.63%-ASR.MA.LL	1.65%-sASR.JA.ASR
2.67%-sASR.GE.LL	1.74%-sASR.JA.LL
2.69%-ASR.GE	1.79%-sASR.SP.LL
2.69%-ASR.GE.ASR	1.82%-ASR.sGE
2.69%-ASR.MA	1.82%-ASR.sGE.ASR
2.69%-ASR.MA.ASR	1.82%-J-EG.sASR
2.69%-J-SP.sASR.LL	1.82%-J-EG.sASR.ASR
2.72%-J-MA.sASR	1.82%-J-MA.sASR
2.72%-J-MA.sASR.ASR	1.82%-J-MA.sASR.ASR
2.74%-J-EG.ASR.LL	1.82%-J-MA.sASR.LL
2.76%-sASR.GE	1.84%-ASR.sMA

all (approximately 1200) variations of predictor and predicted streams. This brute force search is responsible for the major improvement. It allowed us to find much better combinations of streams.

In the table 9 and 10 we present Equal Error Rates for various combinations of predictor and predicted streams. Each column of the table corresponds to a distinct number of training conversations as denoted in its headline (16 conversation training is excluded because of the lack of space). The respective systems are named as Y.X.TRIGGER where the trigger is optional (and filters out frames for which the TRIGGER stream is non-silence). Y is the name of the predicted stream, while X is the predictor stream. Leading 's' in the name of the stream means that the stream was preprocessed by a split-phone tagger (marking phones according to their length). Leading 'J-' in the name of the system means that the joint probabilities of (Y, X) were used, instead of Y|X conditioning. Names EG, SP, GE, MA, JA refer to the open-loop phone streams, LL is the Lincoln-Lab speech activity detector, ASR refers to 30% WER ASR phoneme stream (SRI system). For example sGE.ASR.LL denotes a stream of tagged German phones conditioned by ASR phonemes both of them triggered by the Lincoln-Lab speech activity detector.

We have fused the best systems from these tables by simple addition of score values of the respective systems. We have produced 5 systems, each optimized for its training condition. For example the system ws02-n-ASR-8 is a fusion of the leading systems from the table 10 trained at eight conversations. The systems were added in the order given by their ERR as long as the resulting ERR was decreasing. The fusion method is certainly not the best one, we used it here, just to see what can be reached. The fusion algorithm used approximately top 5 systems from the above tables on average to reach the performance reported in tables 11 and 12. The DET plots for the best of the fused systems can be found in the figure 2.

Table 11: *EERs of fusion, part 1*

1	2
8.37%-ws02-n-ASR-1	3.97%-ws02-n-ASR-2
9.76%-ws02-n-ASR-2	4.95%-ws02-n-ASR-1
17.62%-ws02-n-ASR-4	5.63%-ws02-n-ASR-4
19.43%-ws02-n-ASR-8	6.36%-ws02-n-ASR-8
26.45%-ws02-n-ASR-16	10.58%-ws02-n-ASR-16

Table 12: *EERs of fusion, part 2*

4	8
1.71%-ws02-n-ASR-8	0.97%-ws02-n-ASR-8
1.89%-ws02-n-ASR-4	1.13%-ws02-n-ASR-16
2.11%-ws02-n-ASR-2	1.24%-ws02-n-ASR-4
2.74%-ws02-n-ASR-1	1.67%-ws02-n-ASR-2
3.01%-ws02-n-ASR-16	2.65%-ws02-n-ASR-1

14. Discussion

The method is able to reach EER below one percent at 8 conversation training. On the other hand one should not be too optimistic, because ASR system we used was trained on SWB-1 too and PPRLM open-loop recognizers were using external gender information. So the true performance could be lower.

In comparison, GMM-UBM [6] baseline we were using at WS02 achieved 0.65% EER using 8 conversation training. While the GMM-UBM is better and doubtlessly faster (since it does not require ASR) the method presented here has its advantages. In situations where one performs ASR anyway (and does not mind to pay an extra cost of running the open-loop recognizers) it turns out to be very cheap. Once we have the models trained, the score calculation is a matter of inner product of vectors of dimension say 30000 (for 5 fused streams) which is very fast in comparison with GMM-UBM. So it could be possibly used as a part of the ASR system to trigger a speaker specific language model after first three minutes of his speech, for example. Another application possible, could be finding the N best speaker models out of many possible. ASR is expensive but since the inner product is cheaper than what GMM-UBM usually does, the resulting system could be faster if searching thru a large database. It could be used as a "fast match" for

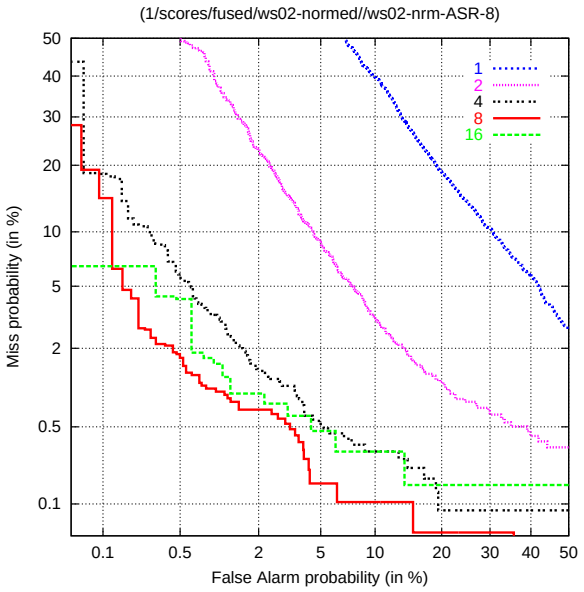

Figure 2: *The DET curves of the fused system ws02-n-ASR-8.*

further refinement by other methods such as GMM-UBM. This "fast match" could even constitute a recursive mean of itself, since we can for example use non-fused streams with non-split phones (inner product of size roughly 1500) to find candidates which could be later pruned by the same method with more complicated streams.

The future work will be aimed to search for a prediction streams that does not require ASR, thus will be faster in one-shot systems. The other direction of research will try to find a system working with ASR alone, not using the open-loop phones. These will be intended for cheap integrated usage within the ASR system since we think that it might be useful for the work of the ASR itself to know who is talking to it.

15. Conclusions

We have shown the new optimal detector especially designed for the case with the lack of the training data. It turns out that this detector is more sensitive to the violation of the assumption of the independence of the features than an ordinary likelihood ratio detector. It can, however, outperform the old detector when special actions such as averaging are done. Unfortunately, we still have no good method of the proper setting of the parameters arising in the averaging process. This is the main reason why this detector could not be used in complicated setup of conditional pronunciation modeling. The tests of this detector were carried out on the "toy"-detector using only English open loop-phones as a features, where the speed of processing allowed us to find the parameters using the brute force.

Further we reported the current status of the conditional pronunciation modeling method, which is still using the old likelihood ratio detector. The future research will concentrate to the possibility of using the new detector in CPM, and to find better features for CPM alone as stated in the previous section.

16. References

[1] D. Klusáček, J. Navrátil, D. Reynolds, J. Campbell, "Conditional Pronunciation Modeling in Speaker Detection" In *ICASSP 2003 Proceedings*, volume IV, pages 804–807, 2003

[2] D. Klusáček, "Pronunciation Modeling in Speaker Detection – Final Report.", JHU-CLSP, `http://atrey.karlin.mff.cuni.cz/~klusacek/jhu_final`, 2003

[3] J. P. Campbell, "Speaker Recognition: A Tutorial", In *Proceedings of the IEEE*, vol. 85, No.9, September 1997

[4] D. Reynolds, "SuperSID Final Presentation" `http://www.clsp.jhu.edu/ws2002/groups/supersid/supersid-final.pdf`, 2002

[5] D. A. Reynolds, et al., "The SuperSID Project: Exploiting High-level Information for High-accuracy Speaker Recognition," accompanying paper in Proc. of the ICASSP-2003

[6] D. Reynolds, T. Quatieri and R. Dunn, "Speaker Verification Using Adapted Mixture Models," In *Digital Signal Processing*, Vol. 10, pp. 181-202, 2000