# On calibration of language recognition scores

*Niko Brümmer*

Spescom Datavoice
Stellenbosch, South Africa.
nbrummer@za.spescom.com

*David A. van Leeuwen*

TNO Human Factors
Soesterberg, the Netherlands
david.vanleeuwen@tno.nl

## Abstract

Recent publications have examined the topic of calibration of confidence scores in the field of (binary-hypothesis) speaker detection. We extend this topic to the case of multiple-hypothesis language recognition. We analyze the structure of multiple-hypothesis recognition problems to show that any such problem subsumes a multitude of derived sub-problems and that therefore the calibration of all of these problems are interrelated. We propose a simple global calibration metric that can be generally applied to a multiple-hypothesis problem and then demonstrate experimentally on some NIST-LRE-'05 data how this relates to the calibration of some of the derived binary-hypotheses sub-problems.

## 1. Introduction: What is calibration?

There has been much recent interest in the topic of *calibration* of speaker detection confidence measures [1, 2, 8, 9, 10, 6]. This paper extends this topic to the case of language recognition. Calibration in language recognition is qualitatively different, because in language recognition there are multiple instead of just two hypotheses.

The issue of calibration of language recognition scores has been addressed in the NIST Language Recognition Evaluations (LRE's) [5], via the pooling (over all target languages) of one-against-the-rest detection scores. The calibration of these pooled scores were then analyzed with the same tools (DET-curves, EER and '$\min C_{\mathrm{DET}}$') that are familiar in the NIST Speaker Recognition Evaluations (SREs) [6]. However in discussions and presentations at the December 2005 LRE Workshop, it became clear that there are some problems associated with the analysis of pooled scores. Briefly, all of these analysis methods assume the use of a single decision threshold, but there cannot be a single threshold that is valid for the pooled scores. This paper is intended to be a constructive response to this analysis problem. In summary, we propose two alternate calibration analyses. One is simply to keep scores for different targets separate and to analyze them separately. The other involves a global calibration transformation of the relative likelihoods of all the languages.

We introduce the topic by giving an intuitive definition of calibration. The purpose of speech processing technology is to extract relevant information from speech. If the technology is good, then this information should enable the user to derive benefit from employing it. In general, the 'better' the quality of this information, the more benefit can be derived.

There are many different ways to measure quality of information. Indirect measurements judge the benefit derived from using information in specific applications. The most well-known indirect measure of information is to employ the information to make recognition decisions (such as *who is speaking*, *what is being said*, *in what language*) and then to estimate error-rates.

It is also possible to directly measure the empirical amount of information, in *bits of Shannon entropy*, that a given speech technology delivers to the user in a set of supervised recognition trials. In fact, as we have pointed out in previous work, there is a very direct relationship between error-rates and information. The information delivered to the user can be expressed as a *total error-rate*, obtained when integrating the average error-rate of a recognizer over a wide range of operating points [2].

In this paper, we shall perform an analysis of the information flow through a language recognition system. Most importantly, we want to be able to measure the amount of information that is delivered to the user by the recognizer. This measurement is not so difficult — all you need is a supervised NIST Evaluation database and equation 17.

But having achieved this, we want to further our analysis to help us to improve the information delivery. We want to decompose our information measure into two components. These components address two important issues:

**Content:** The information must be *there*[1]. The result delivered to the user must actually contain the information that we are interested in. In [2] we used the term *discrimination* for a (direct) measure of the information contents. Other authors use the term *refinement* [3, 8, 4]. Well-known indirect measures of information contents include error-rate measures such as equal-error-rate (EER) and cost-based measures such as '$\min C_{\mathrm{DET}}$' as used in the NIST Speaker Recognition Evaluations [6].

**Form:** The information must be in a standard form that is easily interpretable by the user. The user should be able to directly employ the information in standard ways, without needing further knowledge specific to the properties of the recognizer. Even if the information is present, if the user *misunderstands* the information, the information cannot be employed to the user's benefit. This quality of *form* of the result delivered to the user is termed *calibration* [3, 2, 8, 4].

In summary, the decomposition we want to perform is: *information delivered to user = information present − information lost via misinterpretation*.

[1] If it is not there in the first place, no further interpretation of the result can extract more information. This can be formally expressed via the *data processing inequality*.

As noted above, the left-hand side is easily computed. Given this, we need only compute one of the two terms on the right-hand side. The second term on the right-hand side is what we call *calibration loss*. If a recognizer is badly calibrated the loss is high. If it is well-calibrated the loss is close to zero.

As demonstrated in [2], this decomposition can be performed in a well-defined way for the case of a binary-hypothesis recognizer (such as a speaker detector). This decomposition of [2] has been recently adopted as a confidence score analysis tool in the NIST Speaker Recognition Evaluations [6].

But in the case of a language recognizer, where decisions must be made between multiple hypotheses, it becomes much more difficult to define exactly what we mean by the information being 'present' in a way that allows practical measurement. This is the main question addressed in this paper. Much of the difficulty lies in the considerable complexity that arises when generalizing from two to multiple recognition hypotheses. Before specifically addressing the question of calibration, we need to analyze in the next few sections the structure of multiple-hypothesis recognition problems.

## 2. Language recognition decision theory

We analyze how to make language recognition decisions, based on the assumption that decisions are made to minimize the expected cost of decisions. Such decisions are known as Bayes decisions. In order to simplify our exposition, we assume that all types of errors have a cost of one and all correct decisions have costs of zero. In this case, minimum-expected-cost decisions are also *minimum-probability-of-error* decisions, which are also equivalent to *maximum-a-posteriori* (MAP) decisions.

We work within the following framework that is intended to subsume various flavours of recognition problems, which include *open-set* versus *closed-set* and also *detection* versus *identification*.

### 2.1. What is given

The input to a language recognition *trial* is:

- A speech segment $x$.
- A set of $N$ mutually exclusive and exhaustive hypotheses, $\mathcal{H}_N = \{H_1, H_2, \ldots, H_N\}$, about the language of segment $x$.
- A *prior* probability distribution $\Pi_N = (\pi_1, \pi_2, \ldots, \pi_N)$, which quantifies the uncertainty about which hypothesis is true of the language of $x$.

Note well, that we take the prior as *given*. The information conveyed by the prior cannot be extracted by the recognizer from the speech $x$, nor can it be learned from databases of development data. The prior cannot be supplied by the technology — it is dependent on the application where the recognizer is to be employed.

If it is a *closed-set* problem, then each hypothesis will correspond to a single explicitly specified language. If it is an open-set problem, then $N-1$ of the hypotheses each correspond to a single specified language, but the $N$th hypothesis allows for the possibility that $x$ can be in any other (unspecified) language. This open/closed distinction is very important in the design of some of the sub-stages of a language recognition system. But in what follows we shall work only with designs of recognizer where the final decision stages are independent of this distinction. In other words, in what follows, we shall concentrate on how to make decisions in the face of uncertainty about $N$ hypotheses and we are not interested anymore in exactly what these hypotheses are.

### 2.2. What is asked

Asking *which of the $N$ hypotheses in $\mathcal{H}_N$ is true* is just one way of employing a language recognizer. In order to explore other uses of a recognizer, we have to consider compound hypotheses which can be derived from the original hypotheses via *disjunction* and *negation*. For example we can form the two new hypotheses:

$$H_{1\vee 2} = H_1 \text{ or } H_2 \quad (1)$$

$$\neg H_{1\vee 2} = H_3 \text{ or } H_4 \text{ or } \cdots \text{ or } H_N \quad (2)$$

where $\vee$ denotes logical **or** and $\neg$ denotes logical **not**. Now we can form recognition questions:

> A *recognition question* is defined by a derived set, $\mathcal{H}_{M|N}$, of $M$ hypotheses which are disjunctions of the elements of the original set $\mathcal{H}_N$; and where the members of $\mathcal{H}_{M|N}$ are also exhaustive and mutually exclusive[2]. The question is now simply: "Which one of the $M$ hypotheses is true?".

According to this definition[3], $2 \leq M \leq N$. Further discussion and examples follow below.

### 2.3. Problem definition

When we combine what is given and what is asked, this defines a recognition problem. In what follows, we shall work with a given fixed $\mathcal{H}_N$, but we will allow the prior and the set of derived hypotheses to vary.

> We therefore define[4] a *recognition problem* via the pair $(\mathcal{H}_{M|N}, \Pi_N)$.

Since one or more components of the prior can be zero, this description allows us to express all of the recognition problems defined on all of the non-empty subsets of $\mathcal{H}_N$.

#### *2.3.1. Taxonomy*

The form of $\mathcal{H}_{M|N}$ and $\Pi_N$ allows us to define a taxonomy of problem-types:

- When $\mathcal{H}_{M|N} = \{H_t, \neg H_t\}$, we call it a *detection* question, where $H_t \in \mathcal{H}_N$ is the designated *target* hypothesis. In this case we are detecting the target language against all the other languages allowed by $\mathcal{H}_N$. Both the open and closed-set tasks in the NIST-LRE are detection tasks.
- When all but two of the components of $\Pi_N$ are zero, so that say $\pi_i = 1 - \pi_j$, we effectively get $\mathcal{H}_{M|N} = \{H_i, H_j\}$ which is a *pair-wise binary classification* problem.
- When, more generally, $M = 2$, and when *both* of the derived hypotheses can be disjunctions of more than one of the original hypotheses, then we call it a *binary classification* problem. Example: $\mathcal{H}_{M|N} = \{H_{1\vee 2}, \neg H_{1\vee 2}\}$.

[2] The disjunction (or-ing together) of all the hypotheses is always true, while the conjunction (and) of any two different hypotheses is always false.

[3] The case $M = 1$ is vacuous, $\mathcal{H}_{M|N} = \{\mathbf{true}\}$.

[4] A richer class of problem can be defined if costs other than one are allowed for different kinds of errors. But for simplicity here, we weight all errors equally.

- In the special case where $M = N$, and therefore $\mathcal{H}_{M|N} = \mathcal{H}_N$, then we call it an *identification* problem.
- More generally, when $2 < M \leq N$, then we have a *multi-class classification* problem. Example: $\mathcal{H}_{M|N} = \{H_{1\vee 2}, H_{3\vee 4\vee 5}, H_{6\vee 7\vee \ldots \vee N}\}$.

Note that in the case of $N = 2$, the whole taxonomy degenerates, so that there is only one type of problem, namely a binary decision between two simple hypotheses.

### 2.4. Bayes decisions

All of the different kinds of language recognition problems that can be expressed in this framework, can in theory be optimally addressed via Bayes decisions. We shall assume that in a first step, the recognizer maps the speech segment $x$ to a *score*-vector, $\vec{s}$, of low dimensionality. Then we ask, given $\vec{s}$ (and ignoring the original speech $x$), what is the optimal way to make decisions?

We need to compute the posterior probability of each of the $M$ hypotheses and make a maximum-a-posteriori (MAP) decision. To do this, we assume the language recognizer has the means to compute the vector of $N$ relative log-likelihoods:

$$\vec{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} \log p(\vec{s}|H_1) \\ \log p(\vec{s}|H_2) \\ \vdots \\ \log p(\vec{s}|H_N) \end{bmatrix} + \beta \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{3}$$

where the likelihoods can be scaled by an arbitrary common positive scale factor $\mathrm{e}^{\beta}$. Combining likelihoods and the given prior via Bayes' rule, we get the posterior probabilities for the original hypotheses:

$$P_i = P(H_i|\vec{s}) = \frac{\pi_i \mathrm{e}^{\lambda_i}}{\sum_{j=1}^{N} \pi_j \mathrm{e}^{\lambda_j}}, i = 1, 2, \ldots, N \tag{4}$$

where $\lambda_i$ is the $i$th component of $\vec{\lambda}$. Next, we need to compute the posteriors for the derived hypotheses. We simply need to sum probabilities when there are disjunctions and to complement when there are negations. For example:

$$P_{1\vee 2} = P(H_{1\vee 2}|\vec{s}) = P_1 + P_2 \tag{5}$$

$$1 - P_{1\vee 2} = P(\neg H_{1\vee 2}|\vec{s}) = \sum_{i=3}^{N} P_i \tag{6}$$

In summary, there are four steps: (i) compute the likelihoods; (ii) compute the posteriors for all the elements of $\mathcal{H}_N$; (iii) compute the derived posteriors for all the elements of $\mathcal{H}_M$; (iv) choose the maximum derived posterior and output the corresponding hypothesis. Some notes are in order:

- The last three steps are trivial. The whole difficulty lies in computing the scores $\vec{s}$ and the relative likelihoods $\vec{\lambda}$.
- These likelihoods are an application-independent representation of the language information extracted by the speech technology from the speech input. Once we have these likelihoods, we can address any of the above class of recognition problems.
- We need only *relative* values for the likelihoods. This means that the information extracted from the speech by this first recognizer stage is $(N-1)$-dimensional. This is most easily appreciated if we choose $\beta$ such that $\sum \lambda_i = 0$. Now $\vec{\lambda}$ lives in $\mathbb{R}^N$, but this constraint confines $\vec{\lambda}$ to an $(N-1)$-dimensional subspace[5] of $\mathbb{R}^N$, which we denote $\mathcal{L}^{N-1}$. This is just one form in which this information can be represented, there are very many different equivalent $(N-1)$-dimensional ways of representing this information.

We next give detail of how to make MAP decisions in three special cases of interest.

#### *2.4.1. Pair-wise classification*

Let the hypothesis prior have zero components for all but two of the $N$ hypotheses, for example, $\Pi_N = (0, 0, p, 0, 1-p, 0, 0)$. In this case, we can form the log-likelihood-ratio $\lambda_5^3 = \lambda_3 - \lambda_5$ and the threshold $\theta_5^3 = -\log(p) + log(1-p)$. Then the MAP decision rule is:

$$\begin{array}{l} \lambda_5^3 \geq \theta_5^3 \;\mapsto\; \text{recognize } H_3 \\ \lambda_5^3 \leq \theta_5^3 \;\mapsto\; \text{recognize } H_5 \end{array} \tag{7}$$

#### *2.4.2. Identification*

Here $\mathcal{H}_{M|N} = \mathcal{H}_N$ and all $\pi_i$ are non-zero. For any $t, i = 1, 2, \ldots, N$, we can form the log-likelihood-ratio $\lambda_i^t = \lambda_t - \lambda_i$ and the threshold $\theta_i^t = -\log(\pi_t) + \log(\pi_i)$. This then gives the MAP decision rule:

$$\forall i \neq t : \lambda_i^t \geq \theta_i^t \mapsto \text{recognize } H_t \tag{8}$$

That is, we make the decision to recognize $H_t$, *if and only if* all of the $N-1$ log-likelihood-ratios, $\lambda_i^t$, which compare $H_t$ against the other hypotheses exceed their corresponding thresholds. Notice that the identification rule is expressed in terms of the same pair-wise log-likelihood-ratios as used above in pair-wise classifications.

#### *2.4.3. Detection*

For a designated target hypotheses $H_t$, we have $\mathcal{H}_{M|N} = \{H_t, \neg H_t\}$ and all $\pi_i$ are non-zero. We form the *detection log-likelihood-ratio*:

$$\lambda_{\neg t}^t(\vec{\lambda}) = -\log \sum_{i \neq t} \frac{\pi_i}{1 - \pi_t} \mathrm{e}^{-\lambda_i^t} \tag{9}$$

and the threshold $\theta_{\neg t}^t = -\log(\pi_t) + \log(1 - \pi_t)$, which give the decision rule:

$$\begin{array}{l} \lambda_{\neg t}^t(\vec{\lambda}) \geq \theta_{\neg t}^t \;\mapsto\; \text{recognize } H_t \\ \lambda_{\neg t}^t(\vec{\lambda}) \leq \theta_{\neg t}^t \;\mapsto\; \text{recognize } \neg H_t \end{array} \tag{10}$$

Some observations are in order. First, notice that we can still express the detection rule in terms of the pair-wise log-likelihood-ratios. Next, notice that unlike in the above cases, we cannot separate the information provided by the technology ($\vec{\lambda}$) and the information provided by the application ($\Pi_N$) neatly on both sides of the comparisons. To form $\lambda_{\neg t}^t$, the log-likelihood-ratio that compares the target class against the rest, we need to know at least the relative priors between the non-target hypotheses.

Finally, although the detection rule appears very different from the identification rule, they *are* actually closely related. It can be shown that if this detection rule is applied, then $H_t$ is detected *only if*:

$$\forall i \neq t : \lambda_i^t \geq \theta_i^t \tag{11}$$

Exceeding all thresholds is both *necessary and sufficient* for *identifying* $H_t$, but exceeding these same thresholds is merely *necessary* for *detecting* $H_t$.

[5] This subspace, $\mathcal{L}^{N-1}$, is the hyperplane through the origin which is normal to the vector $[1, 1, \ldots, 1]$.

## 3. Measuring quality of $\vec{\lambda}$

Suppose we have a language recognition sub-system, $S(\cdot;\mathcal{H}_N)$ which maps every speech segment $x$ to a log-likelihood-vector: $\vec{\lambda} = S(x;\mathcal{H}_N)$, where $\vec{\lambda} \in \mathcal{L}^{N-1}$. This sub-system $S(\cdot;\mathcal{H}_N)$ is application independent, because it can be used for any of the different types of recognition problem defined by a pair $(\mathcal{H}_{M|N}, \Pi_N)$.

The question that we address in this paper is *how do we measure the quality* of $S(\cdot;\mathcal{H}_N)$? Moreover, can we decompose this measurement into separate components that judge the *content* and *form* of the output $\vec{\lambda}$? In previous work [2], we have addressed this problem in depth for the case $N = 2$. Specifically it was shown that:

- The total information can be measured via a logarithmic *proper scoring rule*.
- The content/form decomposition[6] can be achieved via the *pair-adjacent-violators* (PAV) algorithm.

Here we generalize this work to the case $N > 2$. First, we propose that it is still appropriate to use a more general logarithmic scoring rule for measuring the total information. As is shown in the next sub-section, this presents no difficulty. Thereafter we discuss why it is unfortunately not so straight-forward to form a content/form decomposition when $N > 2$.

### 3.1. Proper scoring rules

Given a probability distribution $\vec{P} = (P_1, P_2, \ldots, P_N)$ for the elements of $\mathcal{H}_N$, a proper scoring rule $\mathrm{R}(\vec{P}, H_t)$, assigns a scalar cost to $\vec{P}$, depending on the hypothesis $H_t$ which is really true. A proper scoring rule R satisfies:

$$\sum_{i=1}^{N} Q_i \,\mathrm{R}(\vec{Q}, h) \leq \sum_{i=1}^{N} Q_i \,\mathrm{R}(\vec{P}, h) \tag{12}$$

where $\vec{P}$ and $\vec{Q} = (Q_1, Q_2, \ldots, Q_N)$ are any probability distributions for $\mathcal{H}_N$. That is, the expectation of the proper scoring rule with respect to $\vec{Q}$ is minimized *if* $\vec{P} = \vec{Q}$. If the inequality is strict, the expected cost is minimized *if and only if* $\vec{P} = \vec{Q}$ and then R is a *strictly proper scoring rule.*

We mention two well-known strictly proper scoring rules:

*3.1.1. Quadratic rule*

$$\mathrm{R_B}(\vec{P}, H_t) = 1 - 2P_t + \sum_{i=1}^{N} P_i^2 \tag{13}$$

This is a generalization of the Brier rule. See [12].

*3.1.2. Logarithmic rule*

$$\mathrm{R_L}(\vec{P}, H_t) = -\log_2 P_t \tag{14}$$

Note that both $\mathrm{R_B}$ and $\mathrm{R_L}$ strictly satisfy condition (12), and that both give non-negative values. But there are important qualitative differences between these rules:

- The range of $\mathrm{R_B}$ is upper-bounded at 2, while that of $\mathrm{R_L}$ is unbounded above. The latter is an *essential* property to have when evaluating the quality of probability distributions that are intended to be generally applied. A probability distribution that asserts $P_t = 0$ when $H_t$ is really true, can lead to *arbitrarily* expensive decisions. This fact should be reflected by the scoring rule.
- The value of $\mathrm{R_L}(\vec{P}, H_t)$ is dependent only on the relevant component $P_t$ and not on the relative values of the other components of $\vec{P}$. Logarithmic scoring rules are unique in this regard [11].

The logarithmic scoring rule has the following appealing interpretation when employed to evaluate the quality of an $N$-ary probability distribution $\vec{P}$. Any probability for a hypothesis in $\mathcal{H}_N$ can be decomposed into a product of two or more probabilities involving disjunctions of hypotheses. The logarithmic scoring rule applied to this product is then the sum of logarithmic rules applied to each of the factors. Therefore, when we evaluate the probability which was given for the true hypotheses, we are also at the same time similarly evaluating the probabilities for all the derived hypotheses which are also true. For example, when $N = 3$, we can write:

$$\begin{aligned} P(H_1) &= P(H_1|\neg H_2)P(\neg H_2) \\ &= P(H_1|\neg H_3)P(\neg H_3) \end{aligned} \tag{15}$$

Then, if $H_1$ is true, then we apply the logarithmic scoring rule to $P(H_1)$:

$$\begin{aligned} -\log_2 P(H_1) &= -\log_2 P(H_1|\neg H_2) - \log_2 P(\neg H_2) \\ &= -\log_2 P(H_1|\neg H_3) - \log_2 P(\neg H_3) \end{aligned} \tag{16}$$

which hows us that in effect we are also applying the logarithmic rule to the two pair-wise classifications $\{H_1, H_2\}$ and $\{H_1, H_3\}$. The rule that evaluates $P(H_1)$ can assign low cost only if the probability distributions involving the implied subproblems also have low cost.

### 3.2. Evaluation via $C_{llr}$

Proper scoring rules evaluate *probability distributions*. But we want to evaluate the *relative likelihoods* $\vec{\lambda}$. The posterior and the likelihoods are related via Bayes' rule (equation 4). When the prior is given this establishes a bijection between these two representations. The problem is that the evaluation by proper scoring rule is dependent on what prior we use. For reasons similar to those in [2], we choose for our purposes a flat prior, namely $\pi_1 = \pi_2 = \cdots = \pi_N = \frac{1}{N}$. The evaluation criterium is now assembled thus:

We are given a set of $T$ supervised evaluation trials, indexed by $t = 1, 2, \ldots, T$. For each trial $t$, we have the relative log-likelihoods $\vec{\lambda}(t) = S(x_t;\mathcal{H}_N)$ as calculated by recognizer for speech segment $x_t$. We also have the true hypothesis $H_t$ for every trial, from which we obtain $\mathcal{I}(H_i)$, denoting the subset of indices $t$ for which $H_t = H_i$. Then:

$$\begin{aligned} C_{llr} &= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\|\mathcal{I}(H_i)\|} \sum_{t\in\mathcal{I}(H_i)} -\log_2 P_i(t) \\ &= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\|\mathcal{I}(H_i)\|} \sum_{t\in\mathcal{I}(H_i)} \log_2 \sum_{j=1}^{N} \mathrm{e}^{-\lambda_j^i(t)} \end{aligned} \tag{17}$$

where $\lambda_j^i(t)$ is the difference between components $i$ and $j$ of $\vec{\lambda}(t)$, $P_i(t)$ is the posterior (equation 4), derived from $\vec{\lambda}(t)$ and the flat prior, and $\|\cdot\|$ denotes set cardinality.

$C_{llr}$ forms an average over trials of the logarithmic scoring rule, where trials have been weighted to synthetically change the prior to be flat. $C_{llr}$ is an empirical measure of information expressed in terms of *bits of Shannon entropy*. $C_{llr}$ has the following properties:

[6] That is, a *refinement/calibration* or *discrimination/calibration* decomposition.

- $0 \leq C_{llr} \leq \infty$
- $C_{llr}$ represents information *loss*. A perfect recognizer (that makes no errors) will have zero loss, while all others have positive loss.
- There is a reference level at $C_{llr} = \log_2 N$. This corresponds to a useless recognizer that outputs $\lambda_j^i(t) = 0$, for every $i$ and $j$. This recognizer gives us no information about the language, but it is still well-calibrated because it acknowledges this lack of information by its zero outputs.
- A badly calibrated recognizer can have $C_{llr} > \log_2 N$. This is an indication that on average, it would be better to make decisions based only on the prior and to not use the recognizer.
- A value of $C_{llr} < \log_2 N$ means the recognizer is useful and $\vec{\lambda}(t)$ can be expected to give Bayes decisions that are better on average than those based on the prior alone.

# 4. Judging calibration

$C_{llr}$ gives a measure of the total quality of the information delivered to the user (after the calibration loss). How do we now isolate the calibration loss?

In the case of $N = 2$, there is a well-defined way of doing this [2]. We ask by how much the quality of the information can be improved by re-calibrating $\vec{\lambda}$ (which is one-dimensional) via an *information-preserving* calibration transformation. An information-preserving transformation in this case is simply an invertible one. For a well-behaved continuous transformation from $\mathbb{R}$ to $\mathbb{R}$ to be invertible, it has to be either strictly monotonically rising or falling. To preserve the sense of $\vec{\lambda}$, we choose the former. This monotonicity constraint can now be applied to the non-parametric (PAV) optimization of a calibration mapping, where the evaluator[7] uses knowledge of the true hypothesis for each trial. After the optimum calibration transformation is applied, $C_{llr}$ can be computed again. This optimized value is denoted as $\min C_{llr}$. The *calibration loss* is then $C_{llr} - \min C_{llr}$.

In the case $N > 2$ we would like to use the same strategy. We can again consider the optimization of an information-preserving calibration transformation. This transformation must certainly be invertible. However, we are now considering a multi-variate transformation from $\mathcal{L}^{N-1}$ to $\mathcal{L}^{N-1}$. In this case, things like monotonicity and 'sense' are not so easy to define, let alone to enforce such constraints on a non-parametric optimization procedure. Also keep in mind that the information in $\vec{\lambda}$ is an interrelated mixture of information streams about the relative likelihoods of all of the possible hypotheses that can be derived from $\mathcal{H}_N$. One could consider restricting the calibration transformation such that these streams remain independent in the calibration transformation. But again, these streams are interrelated and cannot be kept independent easily. Below we derive and employ one such class of transformation which at least keeps the information that discriminates between pairs of simple hypotheses (elements of $\mathcal{H}_N$) independent.

### 4.1. Direction-preserving calibration

It is possible to find an invertible calibration transformation $\mathcal{T}_\mathcal{L} : \mathcal{L}^{N-1} \mapsto \mathcal{L}^{N-1}$, such that $\vec{\lambda}' = \mathcal{T}_\mathcal{L}(\vec{\lambda})$ has the following properties: Let $\lambda_i'$ and $\lambda_j'$ be any two components of $\vec{\lambda}'$; and let $\lambda_i$ and $\lambda_j$ be the corresponding components of $\vec{\lambda}$, then:

- $\lambda_i' - \lambda_j'$ is dependent (via $\mathcal{T}_\mathcal{L}$) only on $\lambda_i - \lambda_j$ and not on any of the other components of $\vec{\lambda}$,
- the transformation from $\lambda_i - \lambda_j$ to $\lambda_i' - \lambda_j'$ is strictly monotonic rising.

The difficulty here is that there are $(N^2 - N)/2$ non-trivial sets of requirements, but $\mathcal{L}^{N-1}$ is only $(N-1)$-dimensional. The most general form of this transformation is:

$$\mathcal{T}_\mathcal{L}(\vec{\lambda}) = \alpha\vec{\lambda} + \vec{\gamma} \tag{18}$$

where $\alpha$ is a positive scalar and $\vec{\gamma} \in \mathcal{L}^{N-1}$. This transformation allows only scaling and translation. These operations[8] preserve the *direction* of $\vec{\lambda}$. The calibration transformation has a total of $N$ independent scalar parameters.

This solution is very different from the monotonicity constraint in the case $N = 2$ and it lacks many of the pleasing properties of the non-parametric PAV solution [2]. In order to choose parameters for this transformation, we propose optimizing the objective $C_{llr}$ of equation 17. This optimization can be performed with *logistic regression*, such as described in [7].

This, therefore is a first answer to finding a measure of calibration:

1. Calculate $C_{llr}$ on the original data, $\vec{\lambda}(t)$. This represents[9] the total quality of the information delivered to the user.
2. Optimize the parameters $\alpha$ and $\vec{\gamma}$ of $\mathcal{T}_\mathcal{L}$ over the evaluation data, then re-calculate $C_{llr}$ on the transformed data $\mathcal{T}_\mathcal{L}(\vec{\lambda}(t))$. This new value is called $C_{llr}^\mathcal{T}$. This is an estimate of information *content*[10] in $\vec{\lambda}(t)$.
3. $C_{llr} - C_{llr}^\mathcal{T}$ is the *calibration loss*, or the information lost to misinterpretation.

This provides us with a single scalar measurement of calibration. However, it is also very instructive to specifically examine the individual calibrations of derived sub-problems. In the next two sections we discuss calibration analyses of *pair-wise* and *one-against-the-rest* recognition.

### 4.2. Pair-wise calibration analysis

The following strategy does not result in a single scalar measure of calibration, nor of a single measure of the information content. Rather, it analyzes separately all of the $\binom{N}{2} = (N^2 - N)/2$ pair-wise binary classifications that can be performed with $\vec{\lambda}$. Detection scores are formed by differences of the components of $\vec{\lambda}$ and then subject to the methods of [2]. Specifically, for each such pair, we calculate (binary) $C_{llr}$ and $\min C_{llr}$, where the latter is computed via PAV.

In our experiments below, we perform this analysis on both raw data $\vec{\lambda}$ and transformed data $\mathcal{T}_\mathcal{L}(\vec{\lambda})$. This is a demonstration of the effects that the $\mathcal{T}_\mathcal{L}$ calibration has on smaller sub-problems.

Note that by design, the calibration transformation $\mathcal{T}_\mathcal{L}$ leaves the information content of all pair-wise log-likelihood-ratios unchanged. (We verified this by calculating the PAV-optimized $\min C_{llr}$ values for each pair, which indeed remain unchanged.)

[7] The evaluator is the party measuring the calibration.

[8] Since $\mathcal{L}^{N-1}$ is a vector-space, it is closed under the operations of addition and scalar multiplication.

[9] Recall $C_{llr}$ represents information loss.

[10] a.k.a. *refinement* or *discrimination*

### 4.3. Detection calibration analysis

Again this strategy does not result in scalar measures. We choose a set of recognition problems from the other end of the complexity scale[11], namely detection problems. Of course, this is also the application chosen by the NIST LREs. We employ the same prior as in NIST LRE-05, namely a prior that is flat over all non-target hypotheses [5]. The detection log-likelihood-ratios $\lambda^t_{\neg t}$ of equation 9 act as binary classification scores, so we can again employ the binary techniques of [2] to calculate $C_{llr}$ and $\min C_{llr}$ and to plot APE-curves.

Note that unlike with the simple case of pair-wise classifications, the information content of the detection log-likelihood-ratios are changed somewhat by the calibration $\mathcal{T}_\mathcal{L}$. This is because the $\mathcal{T}_\mathcal{L}$-calibrated detection log-likelihood-ratios are not functions of the pre-calibrated ones. That is, in general, there is no function $\mathrm{f}(\cdot) : \mathbb{R} \mapsto \mathbb{R}$, so that:

$$\lambda^t_{\neg t}(\mathcal{T}_\mathcal{L}(\vec{\lambda})) = \mathrm{f}(\lambda^t_{\neg t}(\vec{\lambda})) \tag{19}$$

However, we see in our experiments below that the change in information content is very little and that in fact equation 19 is (for this data) closely approximated.

## 5. Experiments

We performed all experiments on the scores produced on the full 30-second test-set of the NIST LRE-05, by the language recognition system TNO-SDV-1 as described in [7]. The hypothesis set $\mathcal{H}_N$ was a closed set of 7-languages, namely English, Hindi, Japanese, Mandarin, Korean, Spanish and Tamil. There were 3578 trials in total. The TNO-SDV-1 system produced a score-vector $\vec{s}$ of dimensionality 149, which was modeled with a Gaussian-back-end to produce a 6-dimensional relative log-likelihood-vector, $\vec{\lambda} \in \mathcal{L}^6$. These scores were subject to the evaluation methods described in this paper.

### 5.1. Calibration analysis with $\mathcal{T}_\mathcal{L}$

The system TNO-SDV-1 obtained a $C_{\mathrm{DET}} = 8.93\%$ (which is an average detection error-rate, see [5] ). Our analysis[12] (with equation 17) gave $C_{llr} = 0.46 \log_2 7$, and after optimizing $\mathcal{T}_\mathcal{L}$, we obtained $C^\mathcal{T}_{llr} = 0.23 \log_2 7$. This is indicative of a large calibration problem. Had our likelihoods been better calibrated, we could have halved $C_{llr}$.

This same calibration also has a significant effect on $C_{\mathrm{DET}}$ for which the value was reduced to $7.03\%$. However, this effect is not as dramatic as the effect on $C_{llr}$. Our detection APE-curves (see figure 2, discussed below) show why — $C_{llr}$ averages accross all target priors, while NIST's $C_{\mathrm{DET}}$ measures only at the specific prior of 0.5. The APE-curves show that for most languages, the error-rate component due to calibration mismatch at 0.5 (logit prior = 0) is not so bad.

The salient feature of the parameters of $\mathcal{T}_\mathcal{L}$ is a scaling factor of 0.38, indicating that our log-likelihood-ratios where somewhat over-confident.

[11] An $N$-ary probability distribution lives in an $(N-1)$-dimensional simplex. Pair-wise problems live along the $\binom{N}{2}$ edges of the simplex. A one-against-the-rest detection problem lives along an interior line segment connecting the target vertex and the opposite face.

[12] Recall that $\log_2 N$ is the reference value for a useless detector.

### 5.2. Pair-wise calibration analysis

We present this analysis via the bar-graph matrix in figure 3 (on a separate page after the references). There are $\binom{7}{2} = 21$ pair-wise comparisons. Note that these comparisons are symmetric[13]. These are values for $C_{llr}$ and $\min C_{llr}$ as obtained when pairs of languages are recognized against each other. For each comparison there are three values depicted by three adjacent bars, as follows:

**left bar** This is $C_{llr}$ as obtained by the unmodified data $\vec{\lambda}$. This reflects the actual performance of the likelihoods. (Note that in the case of English against Hindi, a value of more than 1.0 was obtained.)

**middle bar** This is $C^\mathcal{T}_{llr}$ as obtained after the general calibration of $\vec{\lambda}$ via $\mathcal{T}_\mathcal{L}$. Note that the improvement in all cases is dramatic.

**right bar** This is $\min C_{llr}$ as *individually* optimized via non-parametric PAV optimization, for each language pair. As noted above, $\min C_{llr}$ is valid for both $\vec{\lambda}$ and $\mathcal{T}_\mathcal{L}(\vec{\lambda})$. It would seem that this is an optimization with much more scope for improvement. But we see that the differences between $C^\mathcal{T}_{llr}$ and $\min C_{llr}$ are not so large. This suggests that $\mathcal{T}_\mathcal{L}$ has already fixed most of the calibration mismatch and is therefore a reasonable way to judge calibration.

From this analysis we can also see that English and Hindi were the problem languages, as compared against all the others.

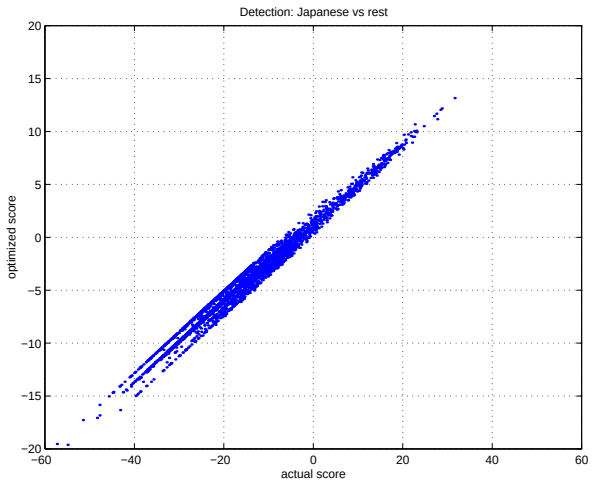



Figure 1: Scatter-plot of original vs optimized scores for detecting Japanese against the rest.

### 5.3. Detection calibration analysis

Refer to figures 2 (a) through (d). These are APE-plots for each[14] of the target languages. There are two APE-plots for each target language. The left plot for each language is for the original detection log-likelihood-ratio, $\lambda^t_{\neg t}$ and the right plot (denoted e.g. `En(opt)`) is for the $\mathcal{T}_\mathcal{L}$-optimized case. Note

[13] These are not confusion error-rates, which are in general not symmetric.

[14] There is no pair-wise recognition here. Each language is detected against all the others. The languages are merely grouped for convenience.

that the scales in different figures are not the same (English and Hindi are still the problem languages).

For a full description of applied-probability-of-error (APE) curves[15], see [2]. Briefly, there are two (solid[16]) curves on an APE-plot, the upper one being the actual error-rate of the detector at a given prior. The lower curve is the error-rate after PAV-optimization. The maximum of the lower curve is the well-known equal-error-rate (EER). The horizontal axis is the prior (in log-odds format). The values of the integrals under the curves are shown as bar-graphs. The integral under the top curve is $C_{llr}$, and is represented by the total height of the bar below the curves. The integral under the bottom curve is $\min C_{llr}$ and is represented by the height of the lower portion of each bar. The area between the two curves is calibration loss, $C_{llr} - \min C_{llr}$ and is represented by the height of the upper portion of each bar. The APE-curve gives a visual appreciation of calibration, but is also a demonstration of the relationship between information and error-rates.

As can be seen from the APE-curves, the pleasing result of this experiment is that the detection calibration, for all seven languages has been much improved. The pre-$\mathcal{T}_{\mathcal{L}}$ detection calibration loss for most of the languages is comparable in magnitude to $C_{llr}$. It is sad to lose so much of the hard-earned information to 'misinterpretation'. But $\mathcal{T}_{\mathcal{L}}$ calibration improves calibration loss for all seven targets to a small fraction of $C_{llr}$.

It is important to note that this experiment does not prove that we can fix the calibration of our language recognition system. The calibration transformation $\mathcal{T}_{\mathcal{L}}$ was optimized over the *supervised* evaluation data itself. We have merely performed a *measurement* of the extent of the calibration problem of our recognition system.

Finally, refer to figure 1. This is a scatter-plot to examine the relationship between pre-calibrated scores, $\lambda^t_{\neg t}(\vec{\lambda})$, on the $x$-axis and post-calibrated scores, $\lambda^t_{\neg t}(\mathcal{T}_{\mathcal{L}}(\vec{\lambda}))$ on the $y$-axis. This example is for Japanese, the others are very similar. Note that the scatter plot indicates that indeed the $y$-axis is strictly speaking not a function (solely) of the $x$-axis. But there is a very strong linear trend, having the a slope of about 0.4, which corresponds to the scaling factor of $\mathcal{T}_{\mathcal{L}}$. This shows equation 19 is indeed approximated in this case.

This observation and the good improvements in detection calibration give more support to the case that $\mathcal{T}_{\mathcal{L}}$ is useful for measuring calibration.

## 6. Conclusion

A language recognizer that is designed to recognize $N$ exhaustive and mutually exclusive language hypotheses via the computation of $N$ relative likelihoods, is also capable of recognizing a wealth of derived hypotheses formed by disjunctions of the original hypotheses. We have examined in detail two of these derived problems, namely pair-wise binary language classification and one-against-the-rest language detection.

We propose an information-theoretic measure, $C_{llr}$ formed by a logarithmic proper scoring rule, for evaluating the quality of these likelihoods. $C_{llr}$ forms an evaluation that is simultaneously applicable to all of the implied sub-problems.

We propose a simple $N$-parameter, information- and direction-preserving, calibration transformation, $\mathcal{T}_{\mathcal{L}}$, the parameters of which can be optimized over supervised evaluation data. This forms a decomposition of $C_{llr}$ that simultaneously gives (i) an estimate of the information content of the uncalibrated likelihoods and (ii) an estimate of the calibration loss incurred because of poor calibration of this information.

We show that auxiliary calibration analyses can be performed with the methods of [2], on sub-problems involving binary decisions between derived hypotheses. We perform these auxiliary analyses on some NIST LRE-05 data to show that the calibration mismatch expressed by $\mathcal{T}_{\mathcal{L}}$ indeed represents a significant part of the calibration loss that is measurable in the binary sub-problems of pair-wise classification and detection.

In summary, we have established an experimentally proven methodology, that provides a practical way of analyzing information flow in $N$-hypothesis recognition problems.

## 7. References


[1] N.Brümmer, "Application-independent evaluation of speaker detection", Odyssey 2004.

[2] N.Brümmer, and J. du Preez, "Application-independent evaluation of speaker detection", Computer Speech & Language, Volume 20, Issues 2-3, April-July 2006, pp 230-275.

[3] M.DeGroot and S.Fienberg, "The comparison and evaluation of forecasters". The Statistician 32, 1222, 1983.

[4] A.Niculescu-Mizil and R.Caruana, "Predicting Good Probabilities With Supervised Learning" in Proc. 22nd International Conference on Machine Learning (ICML), 2005.

[5] Various NIST LRE 'Evaluation Plans' are available at: www.nist.gov/speech/tests/lang/index.htm.

[6] The 2006 NIST SRE Evaluation Plan is available at: www.nist.gov/speech/tests/spk/2006/index.htm.

[7] D. van Leeuwen and N.Brümmer, "Channel-dependent GMM and Multi-class Logistic Regression models for language recognition", Odyssey 2006.

[8] W.M.Campbell et al."Estimating and evaluating confidence for forensic speaker recognition", ICASSP 2005.

[9] W.M.Campbell et al."Understanding Scores in Forensic Speaker Recognition", Odyssey 2006.

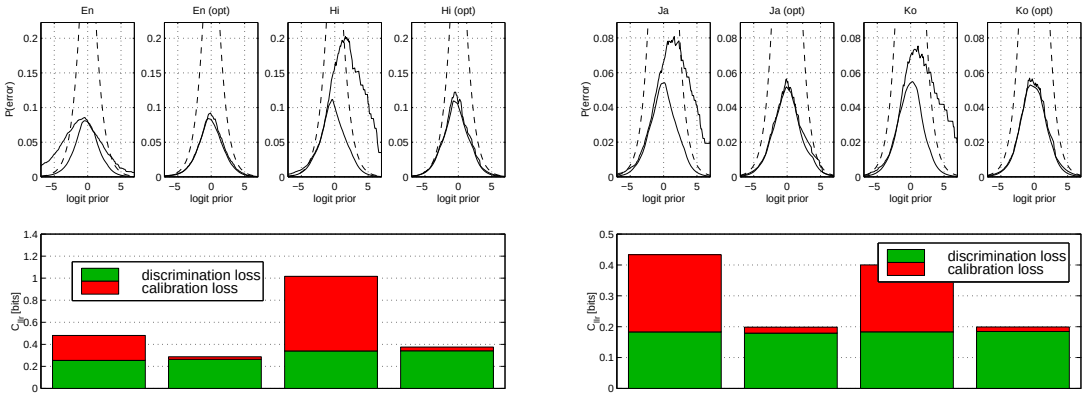
[10] D.Ramos-Castro, J.Gonzalez-Rodriguez and J.Ortega-Garcia, "Likelihood Ratio Calibration in a Transparent and Testable Forensic Speaker Recognition Framework", Odyssey 2006.

[11] N.C.Dalkey, "Inductive Inference and the Maximum Entropy Principle". In: Maximum-Entropy and Bayesian Methods in Inverse Problems, eds.: C.R. Smith and W.T. Grandy, D. Reidel Publishing Company, Dordrecht, pp.351-364, 1985.

[12] P.Sebastiani and H.P.Wynn, "Experimental Design to Maximize Information". MaxEnt 2000: Twentieth International Workshop on Bayesian Inference and Maximum Entropy in Science and Engineering. AIP Conference Proceedings, 2000, pp. 192-203.


[15] MATLAB tools to plot APE-curves are available here: `www.dsp.sun.ac.za/~nbrummer/focal`.

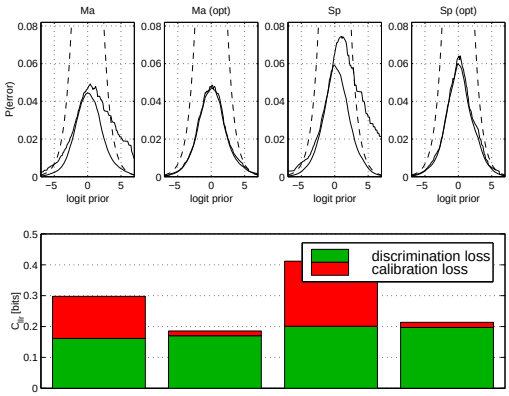
[16] The dashed curve is a reference curve for a useless system having EER=50%.



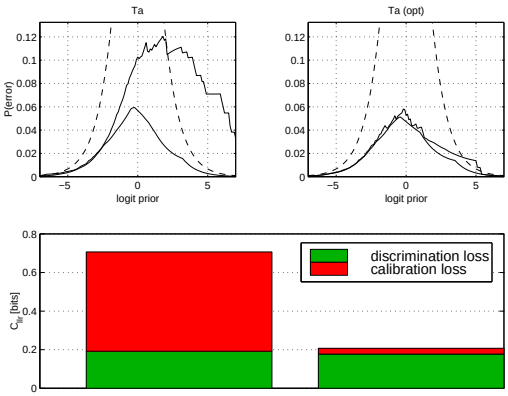

(a) English and Hindi




(b) Japanese and Korean




(c) Mandarin and Spanish




(d) Tamil

Figure 2: Detection Calibration Analysis: Actual vs. $\mathcal{T}_{\mathcal{L}}$-optimized

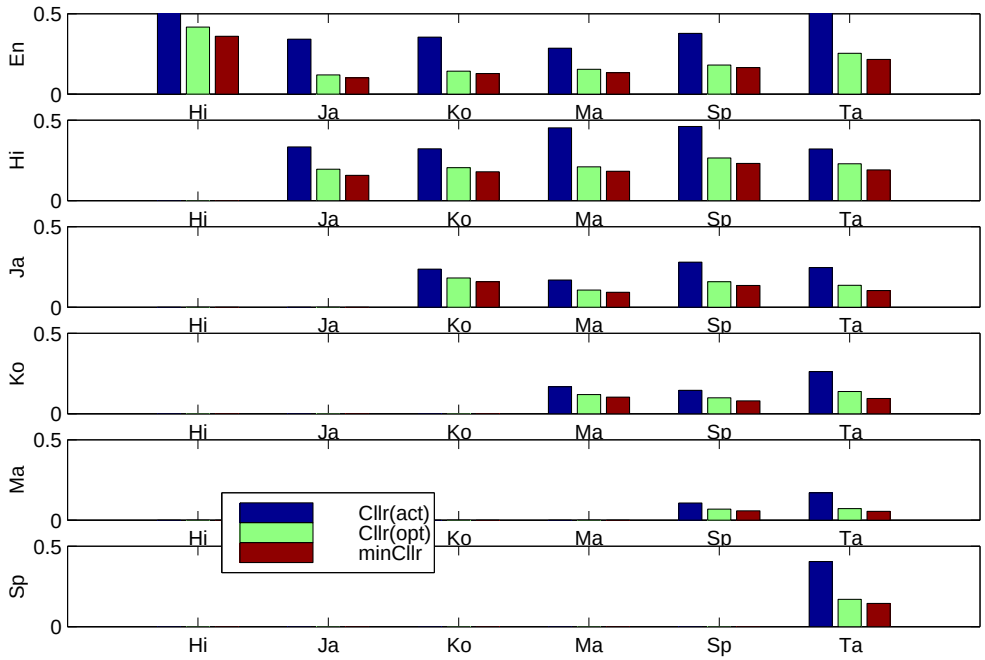



Figure 3: Pair-wise Calibration Analysis: Actual vs. $\mathcal{T}_{\mathcal{L}}$-optimized