Measuring, refining and calibrating speaker and language information extracted from speech

Niko Brümmer

Ph.D. Defence, 18 October 2010
Dept. E&E Engineering. Univ. Stellenbosch
Promoter: Prof. J.A. du Preez

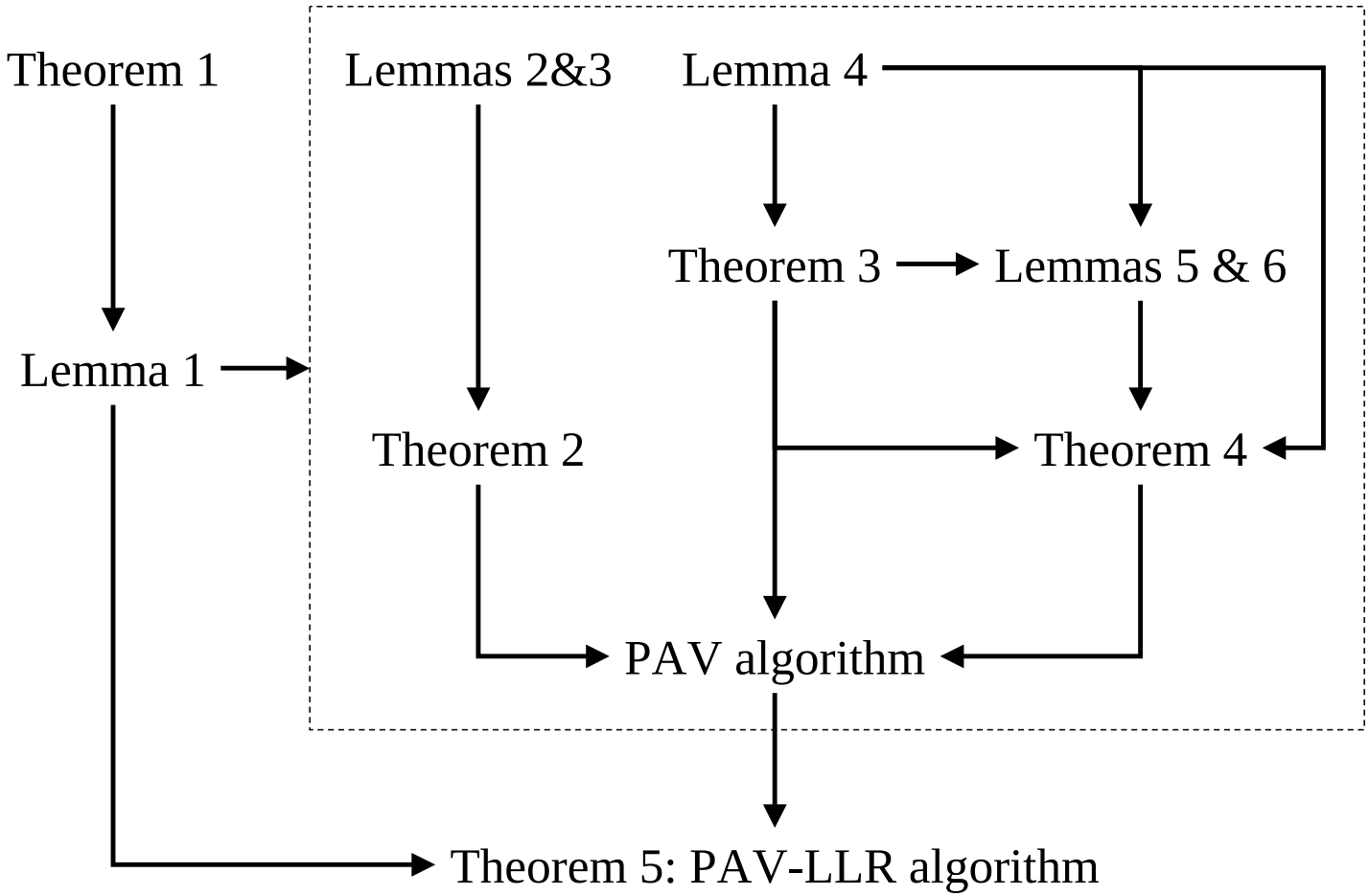
Outline

We can now rewrite the integral (for the case $i = N$) as:

$$
\begin{aligned}
&\int_{\mathbb{P}_N} \Gamma(N) C^*_{\boldsymbol{\eta}}(\mathbf{p}|\theta_N)\,\mathbf{d}\boldsymbol{\eta} \\
&= \int_{-\infty}^{y_1}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \mathcal{I}(\mathbf{x})\,dx_{N-1}\,dx_{N-2}\cdots\,dx_1 \\
&+ \int_{y_1}^{\infty}\int_{-\infty}^{y_2}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \mathcal{I}(\mathbf{x})\,dx_{N-1}\,dx_{N-2}\cdots\,dx_1 \\
&+ \int_{y_1}^{\infty}\int_{y_2}^{\infty}\int_{-\infty}^{y_3}\cdots\int_{-\infty}^{\infty} \mathcal{I}(\mathbf{x})\,dx_{N-1}\,dx_{N-2}\cdots\,dx_1 \\
&+ \cdots \\
&+ \int_{y_1}^{\infty}\int_{y_2}^{\infty}\int_{y_3}^{\infty}\cdots\int_{-\infty}^{y_{N-1}} \mathcal{I}(\mathbf{x})\,dx_{N-1}\,dx_{N-2}\cdots\,dx_1
\end{aligned}
$$

where

$$
\mathcal{I}(\mathbf{x}) = \frac{\Gamma(N-1)}{\eta_N}\prod_{k=1}^{N}\eta_k = \Gamma(N-1)\frac{e^{\sum_{k=1}^{N-1}x_k}}{\left(1+\sum_{k=1}^{N-1}e^{x_k}\right)^{N-1}}
$$


Theorem 1
Lemmas 2&3
Lemma 4
Theorem 3
Lemmas 5 & 6
Lemma 1
Theorem 2
Theorem 4
PAV algorithm
Theorem 5: PAV-LLR algorithm

Introduction

The basic problem

How to evaluate the goodness of a certain class of automatic pattern recognizers.

- Automatic pattern recognizers are not infallible.
- They make errors.
- If one wants to design, improve, sell, buy, or use a pattern recognizer, then it is important to have some understanding of these errors.

In short, the need exists to evaluate the goodness of pattern recognizers.

The pattern recognizers

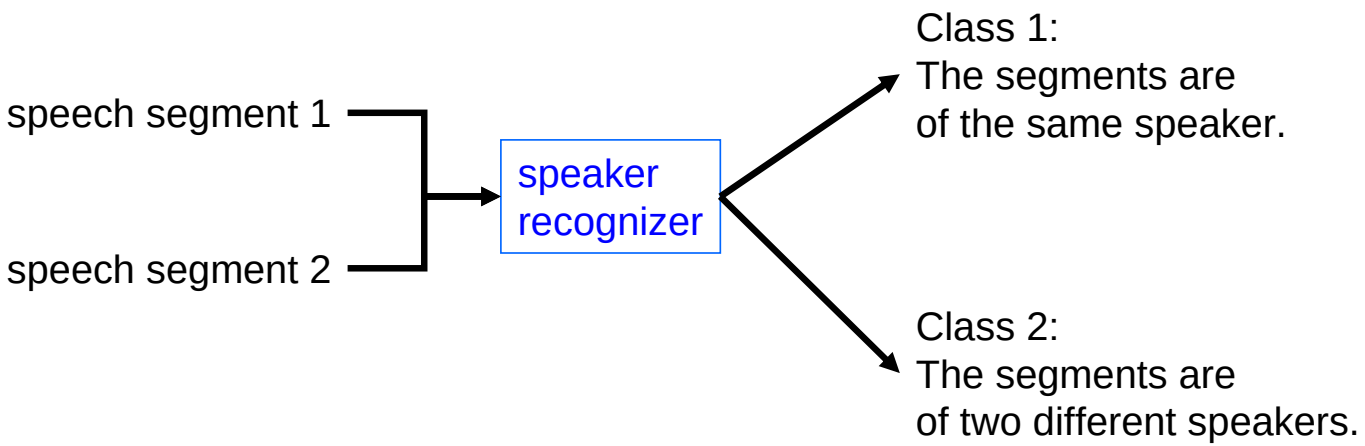
We discuss two kinds of pattern recognizers that automatically extract information from speech:

- Automatic speaker recognition
- Automatic spoken language recognition

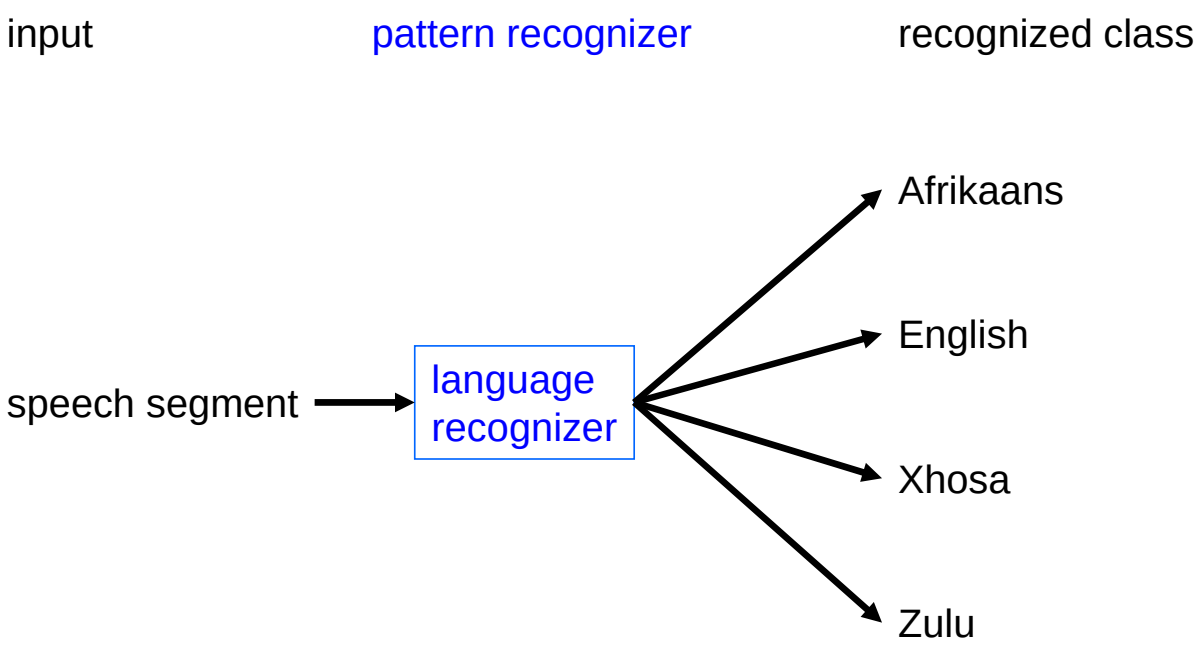
Although this work is grounded in the literature and presented in the terminology of these two fields, it is more generally applicable also to other pattern recognition problems.

The canonical speaker recognizer

input | pattern recognizer | recognized class



Language recognizer



The question is simple

The questions asked of speaker and language recognizers are very simple:

- Is it the same speaker or not?
- Which of these four languages is it?

There is a small discrete number of possibilities.

Compare this to speech recognition, where the question, “What was said?”, is much more complex, because there are very many possibilities.

The answer is not

The answers to these questions are however complicated by the fact that with current technology, they cannot be answered with certainty.

An honest (and therefore more useful) pattern recognizer should reflect the degree of uncertainty in its answer.

How to reflect uncertainty

existing state of the art

For example, a speaker recognizer could output:

- Hard discrete decisions: class 1 or class 2. This is a poor solution, with no indication of uncertainty.
- Scores: more positive scores favour the same-speaker hypothesis, more negative scores favour the different-speaker hypothesis. This is a good solution and is still part of the current state of the art.

But the score is uncalibrated: it is up to the user to exercise the recognizer in order to learn how to interpret the score magnitude as an indication of uncertainty.

How to reflect uncertainty
proposed here

or the speaker recognizer could output various forms of calibrated scores:

- Probability distribution:

$$P(\text{class 1}|\text{speech,prior}), \qquad P(\text{class 2}|\text{speech,prior})$$

- Likelihood distribution:

$$L(\text{class 1}|\text{speech}), \qquad L(\text{class 2}|\text{speech})$$

Likelihood and posterior probability are closely related, but the likelihood (our preferred format), is prior-independent and more useful because it allows user-supplied priors.

Goals

The main goals of this work were:

1. To evaluate the goodness of speaker and language recognizers (or other similar pattern recognizers) that provide outputs in class likelihood format.
2. Given that we can measure goodness of pattern recognizers, how can we improve them?

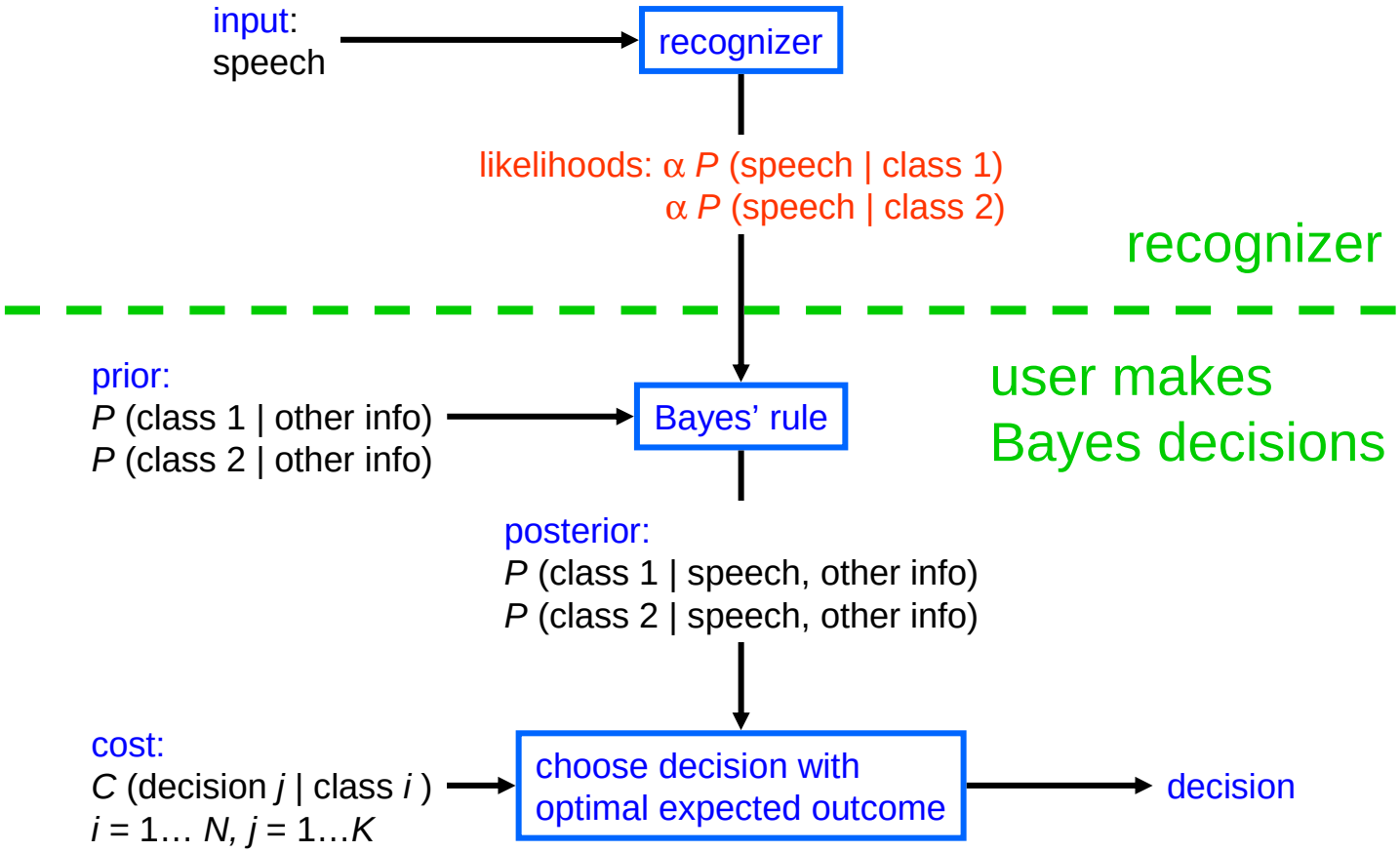
Relevance

- This work builds on the series of NIST Speaker Recognition Evaluations and NIST Language Recognition Evaluations, which have been a major driving force for research in these fields for more than a decade. In the period 2000 to 2010, the author has participated in 7 speaker recognition and 3 language recognition evaluations.
- The practical algorithms developed in this work have been made available in a MATLAB toolkit, the FoCal Toolkit, which has been used by many other researchers, especially for their work in the NIST evaluations.

Theory

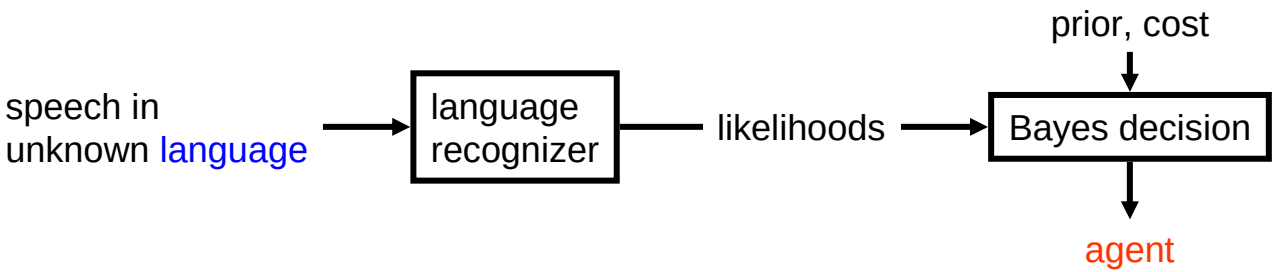
Why likelihoods?

- Likelihoods convey the information extracted from the speech by the recognizer, which the user can employ to make optimal Bayes decisions.
- The information in the likelihoods is application independent. The Bayes decision framework allows the user to apply the likelihoods to a wide range of different applications.


input:
speech
recognizer
likelihoods: α P (speech | class 1)
α P (speech | class 2)
recognizer
prior:
P (class 1 | other info)
P (class 2 | other info)
Bayes' rule
user makes
Bayes decisions
posterior:
P (class 1 | speech, other info)
P (class 2 | speech, other info)
cost:
C (decision j | class i)
i = 1… N, j = 1…K
choose decision with
optimal expected outcome
decision

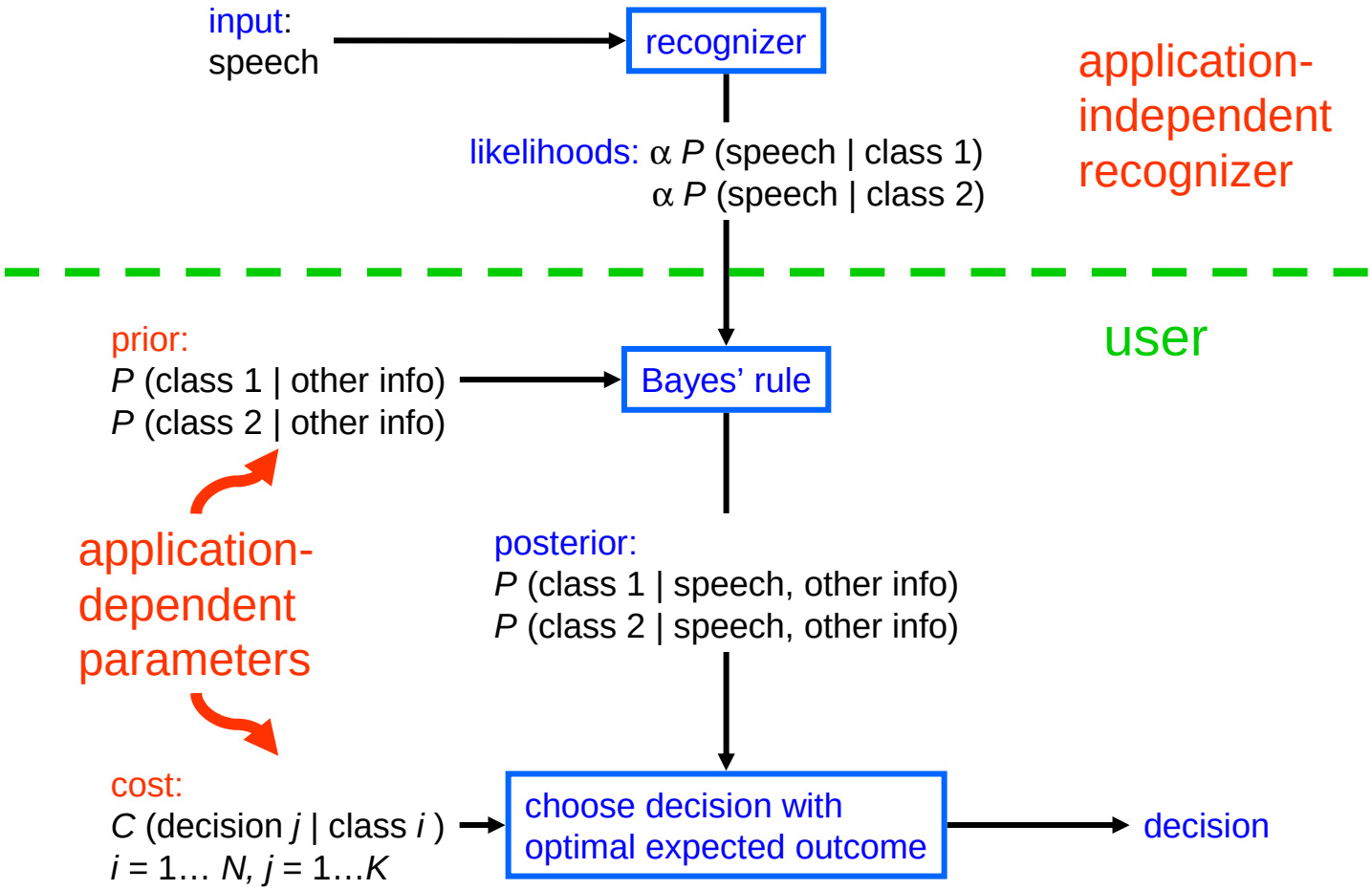
Cost Example

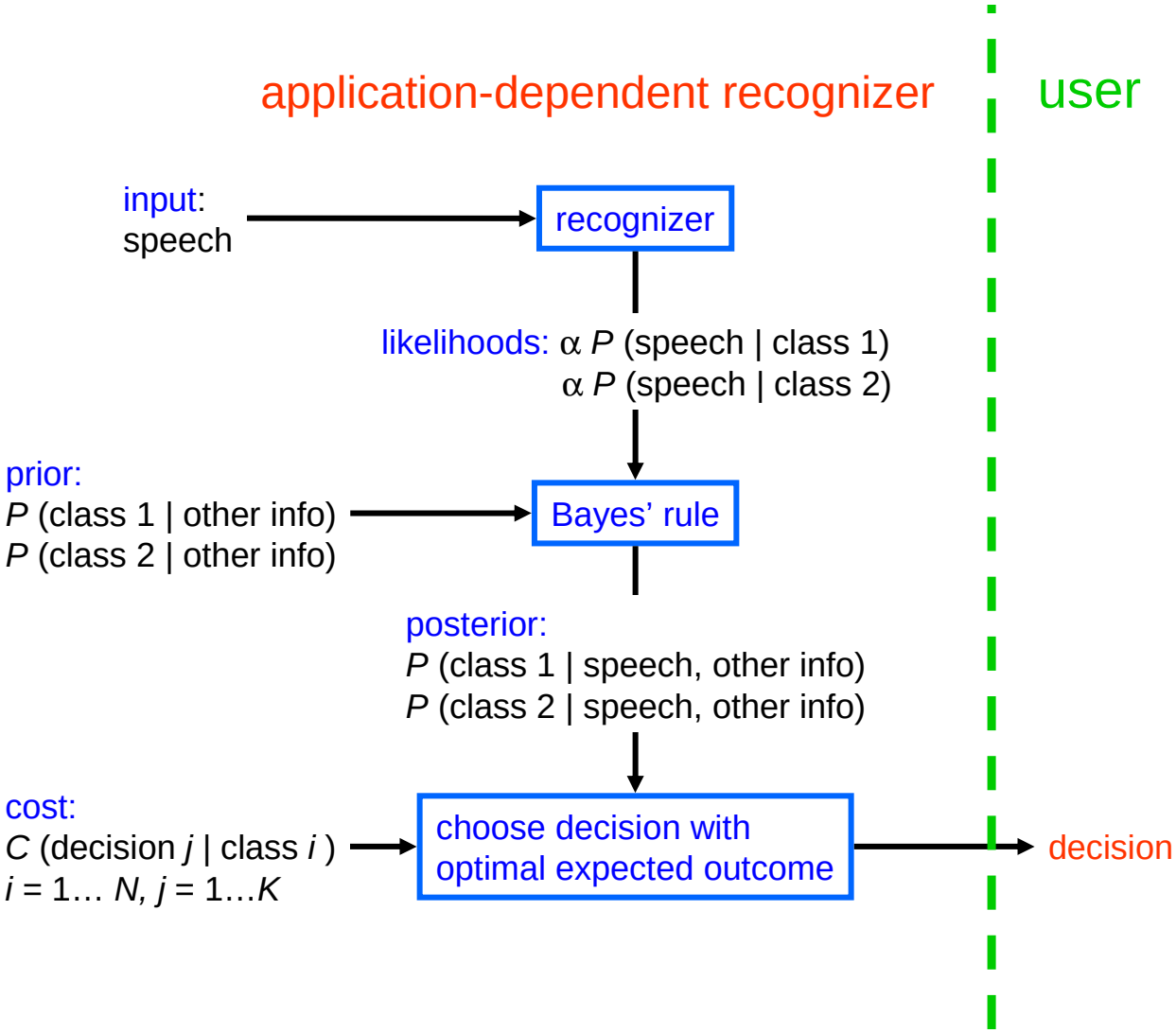
A language recognition application

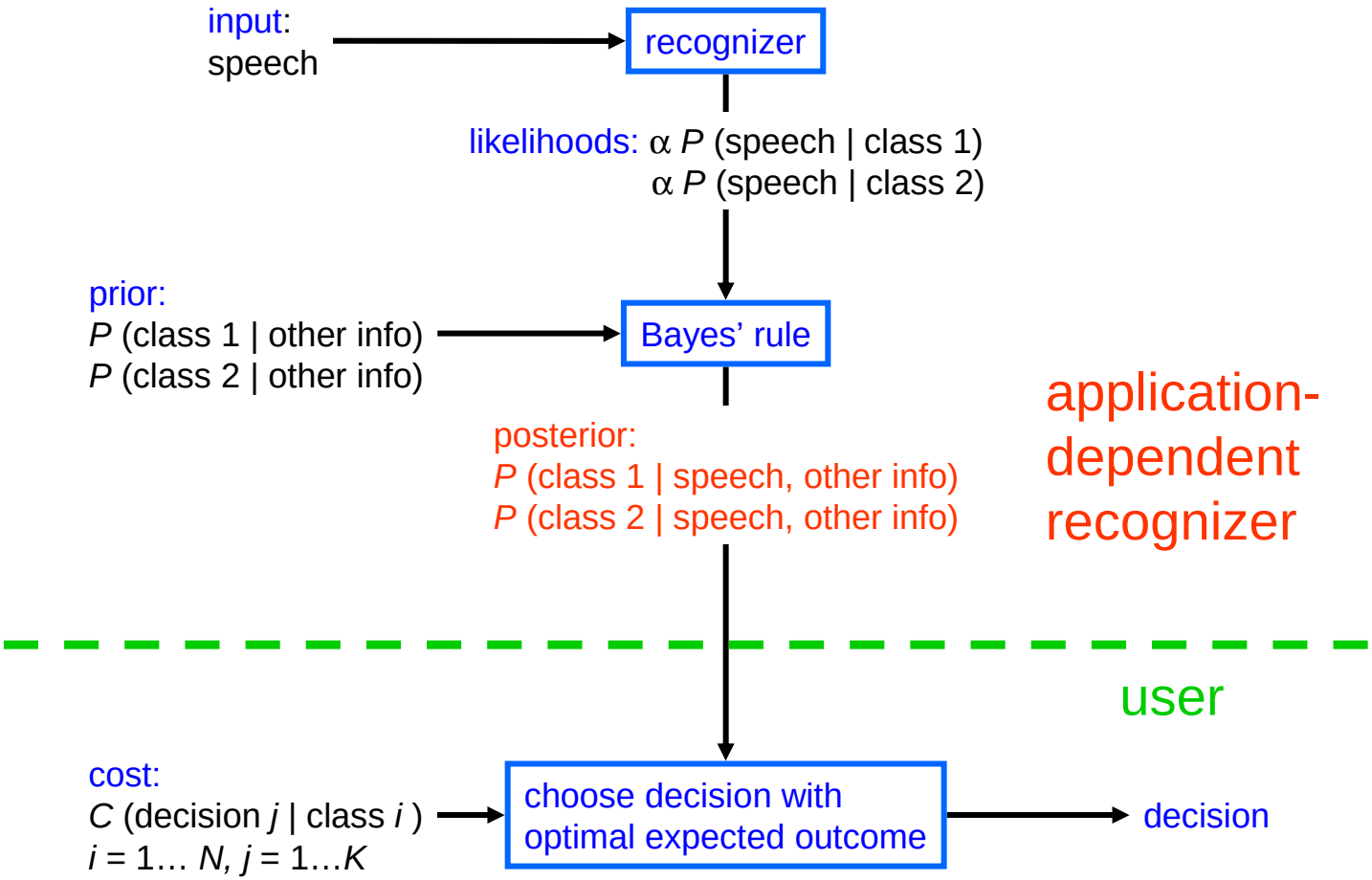


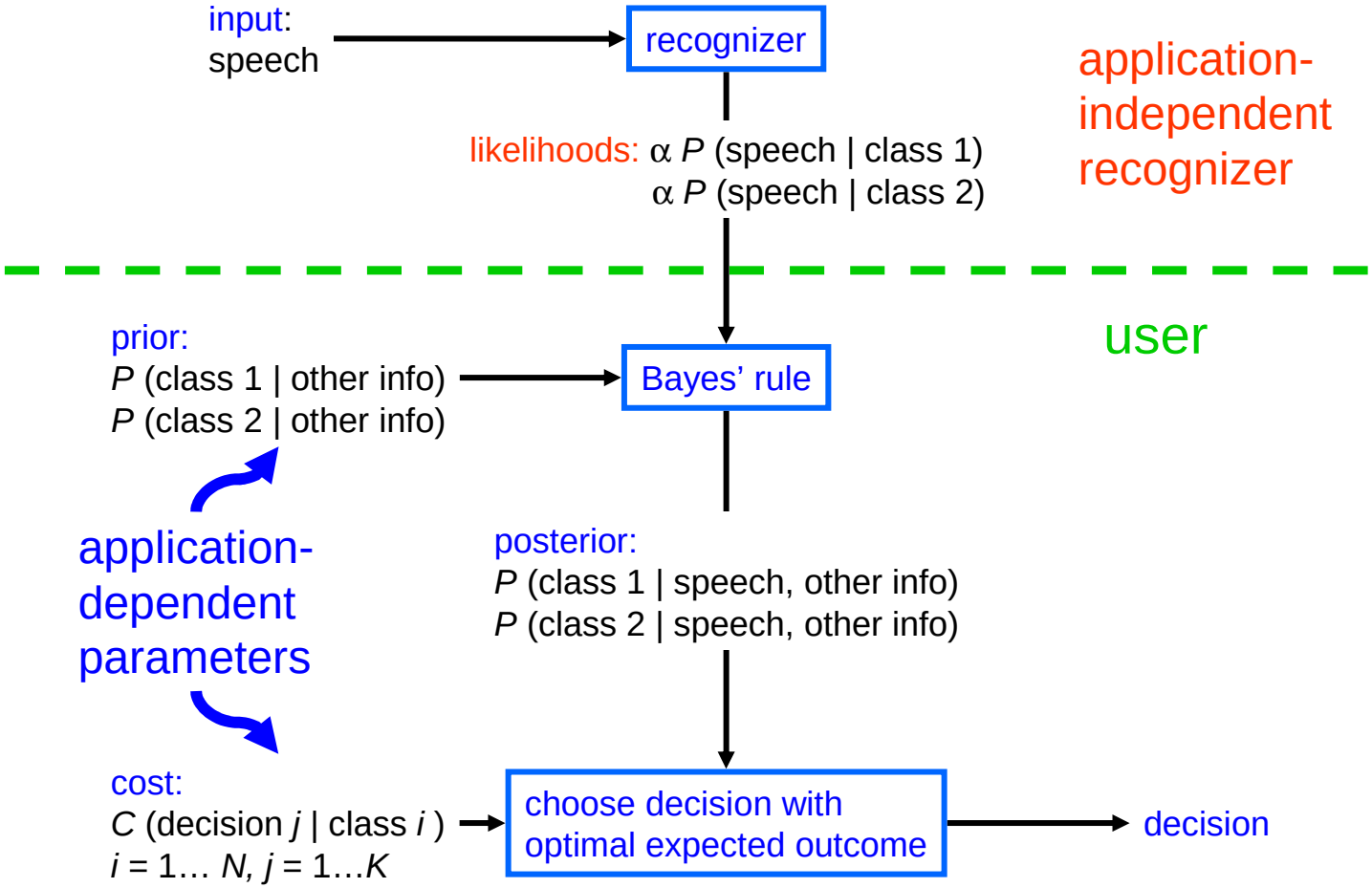
	Afrikaans	English	Xhosa	Zulu
Koos	0	0	2	2
John	1	0	2	2
Nelson	1	0	0	1
Jacob	1	0	1	0

Cost of assigning a speech segment to an agent.


input:
speech
recognizer
likelihoods: α P (speech | class 1)
α P (speech | class 2)
application-independent recognizer
user
prior:
P (class 1 | other info)
P (class 2 | other info)
Bayes' rule
application-dependent parameters
posterior:
P (class 1 | speech, other info)
P (class 2 | speech, other info)
cost:
C (decision j | class i)
i = 1… N, j = 1…K
choose decision with optimal expected outcome
decision


application-dependent recognizer
user
input:
speech
recognizer
likelihoods: α P (speech | class 1)
α P (speech | class 2)
prior:
P (class 1 | other info)
P (class 2 | other info)
Bayes' rule
posterior:
P (class 1 | speech, other info)
P (class 2 | speech, other info)
cost:
C (decision j | class i)
i = 1… N, j = 1…K
choose decision with
optimal expected outcome
decision


input:
speech
recognizer
likelihoods: α P (speech | class 1)
α P (speech | class 2)
prior:
P (class 1 | other info)
P (class 2 | other info)
Bayes' rule
posterior:
P (class 1 | speech, other info)
P (class 2 | speech, other info)
application-dependent recognizer
user
cost:
C (decision j | class i)
i = 1… N, j = 1…K
choose decision with optimal expected outcome
decision


input:
speech
recognizer
application-independent recognizer
likelihoods: α P (speech | class 1)
α P (speech | class 2)
user
prior:
P (class 1 | other info)
P (class 2 | other info)
Bayes' rule
application-dependent parameters
posterior:
P (class 1 | speech, other info)
P (class 2 | speech, other info)
cost:
C (decision j | class i)
i = 1… N, j = 1…K
choose decision with optimal expected outcome
decision

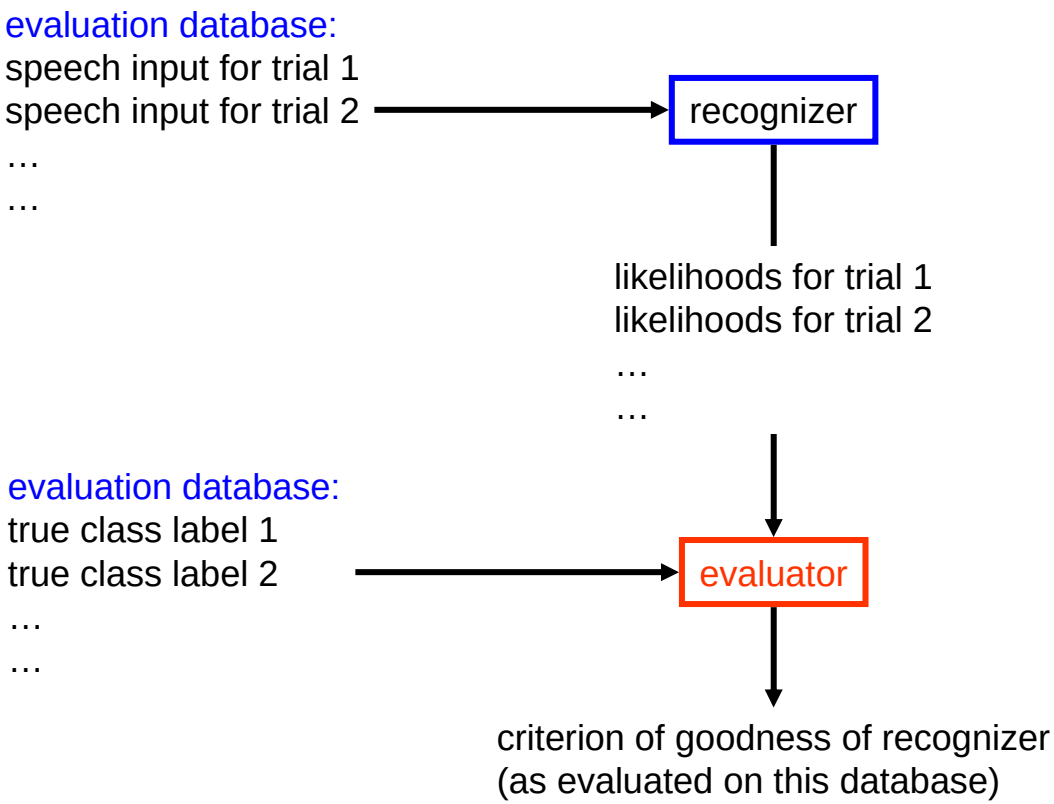
Theory
Evaluation

Recognizer evaluation

To evaluate the goodness of speaker or language recognizers, we follow the paradigm provided by the NIST Speaker Recognition Evaluations and the NIST Language Recognition Evaluations:

- The recognizer under evaluation is exercised on a large database of speech inputs, producing an output in likelihood form for each input.
- The evaluator compares the recognizer's likelihoods with the true class labels to produce a summary of the goodness of the recognizer.

Evaluation via supervised evaluation database



Supervised evaluation of likelihoods

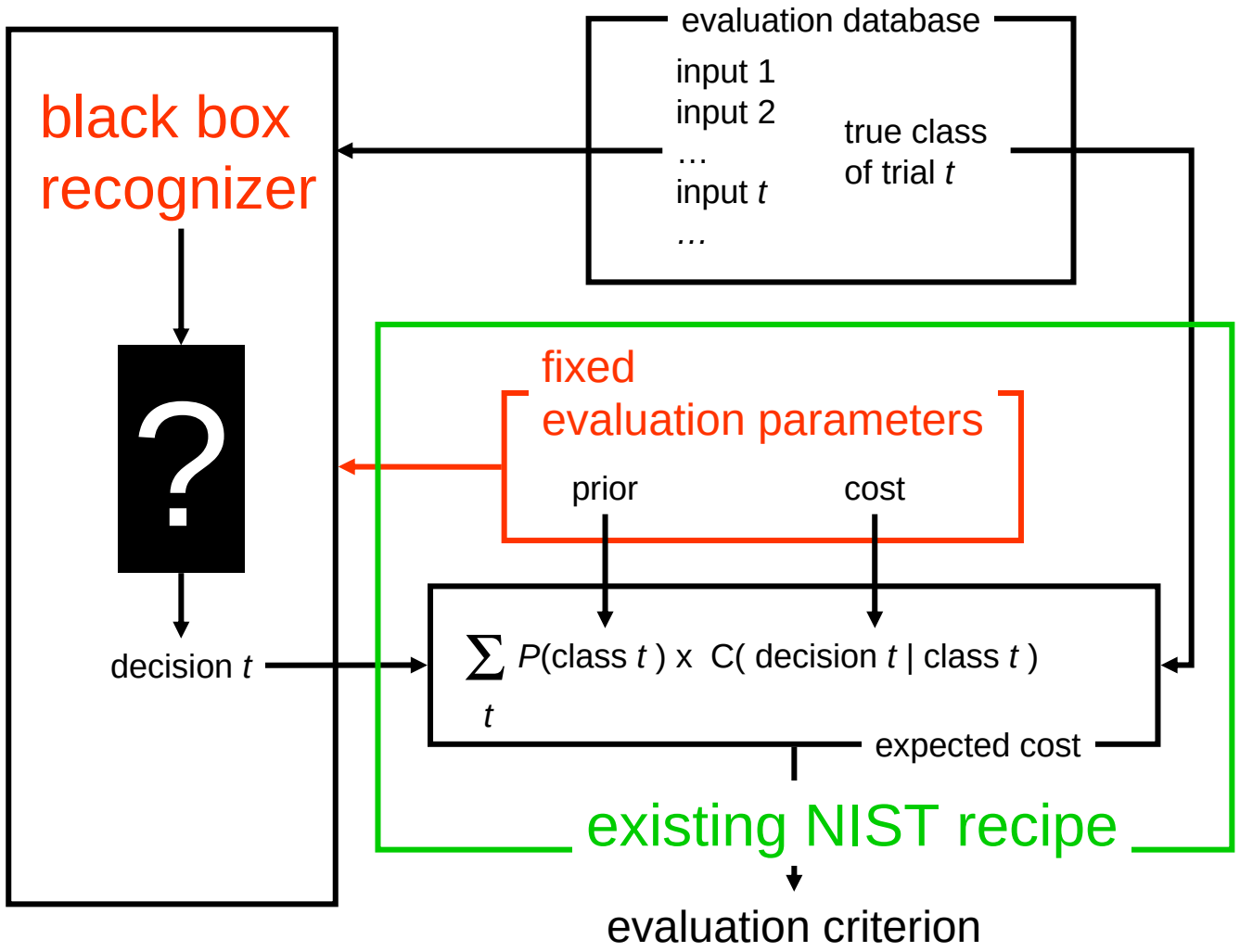
How does the evaluator judge the goodness of the recognizer's likelihoods?

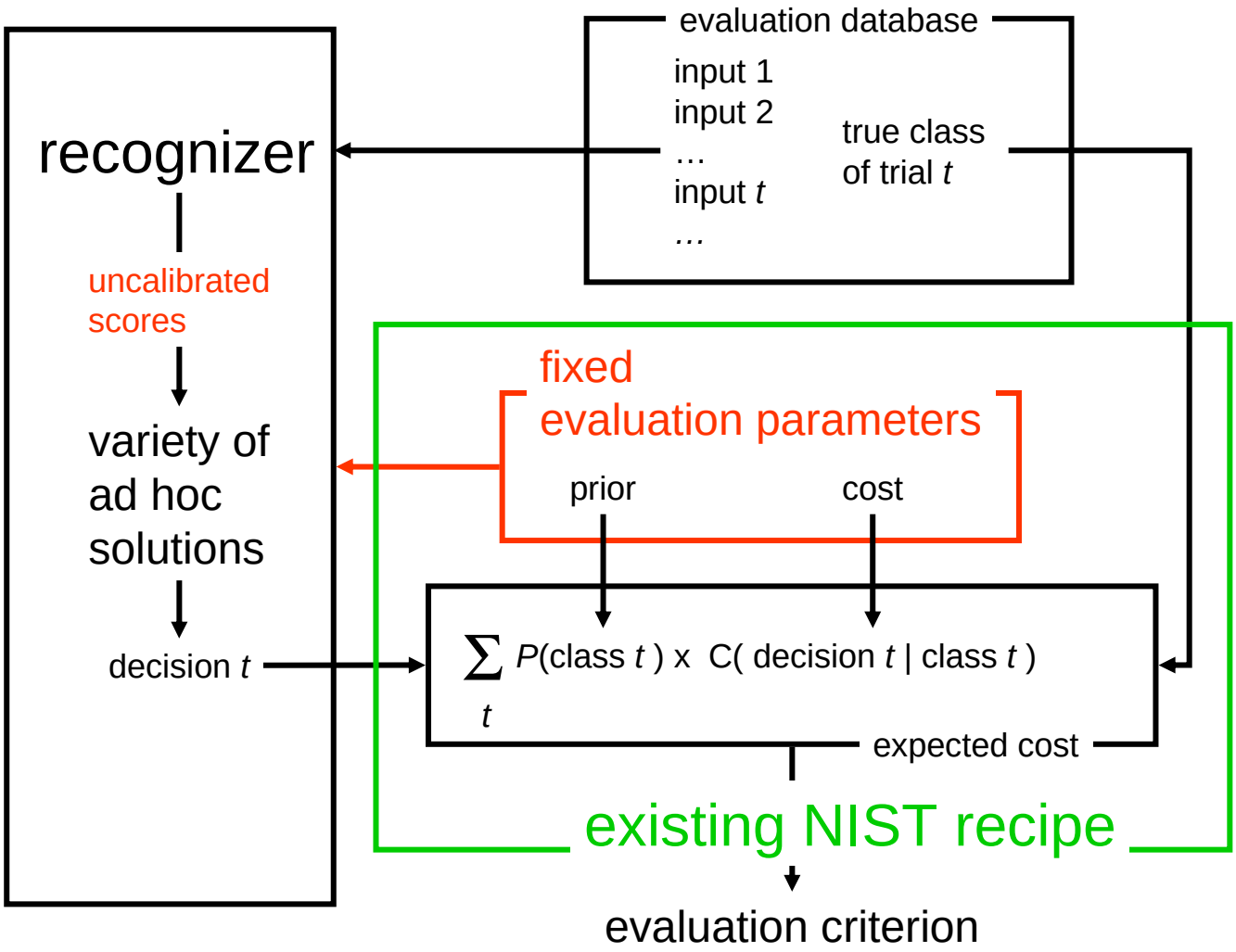
- The evaluator is not given 'true' likelihoods to compare against.
- The evaluator has only the true class labels.

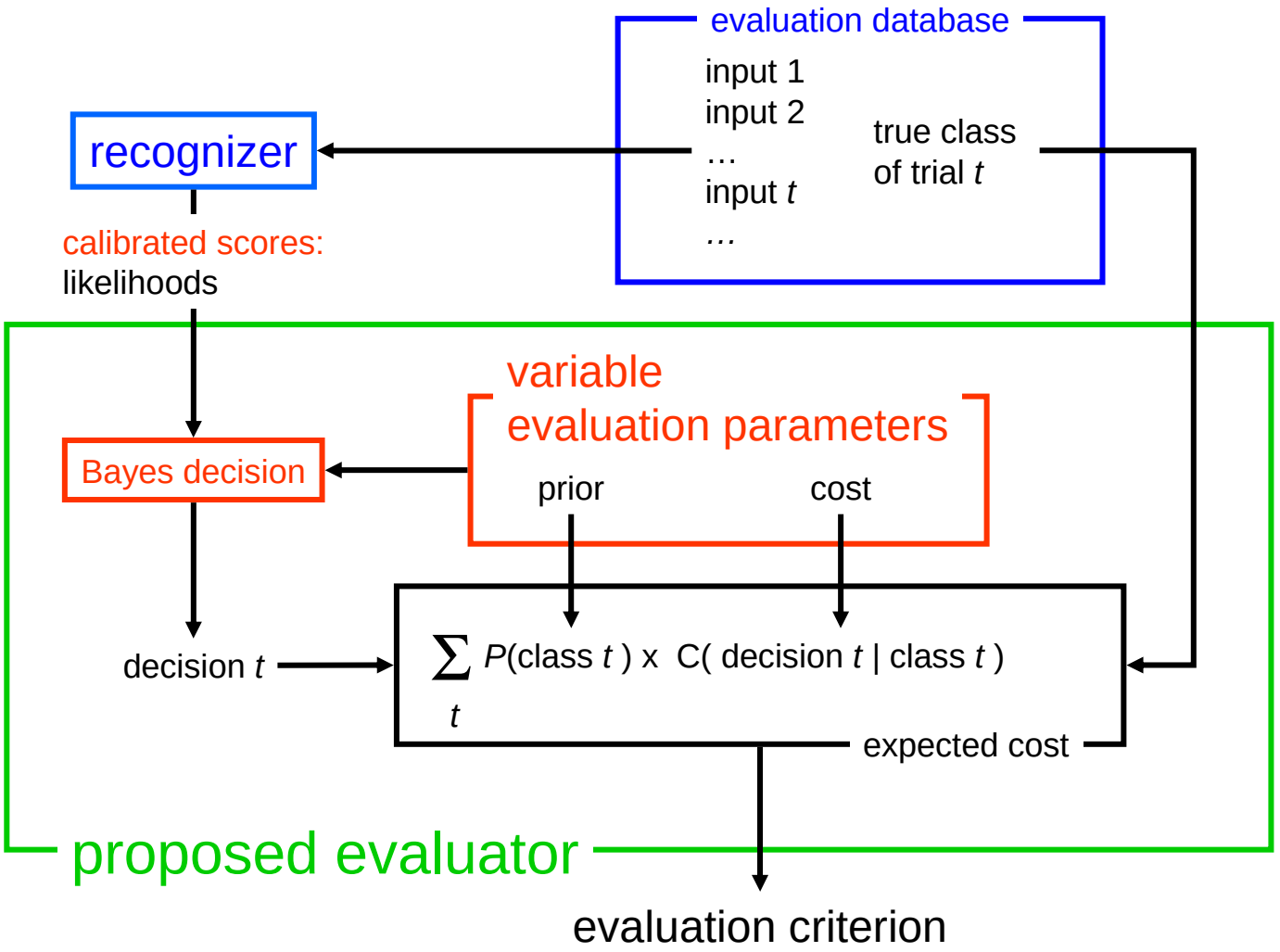
Evaluation by Bayes decision

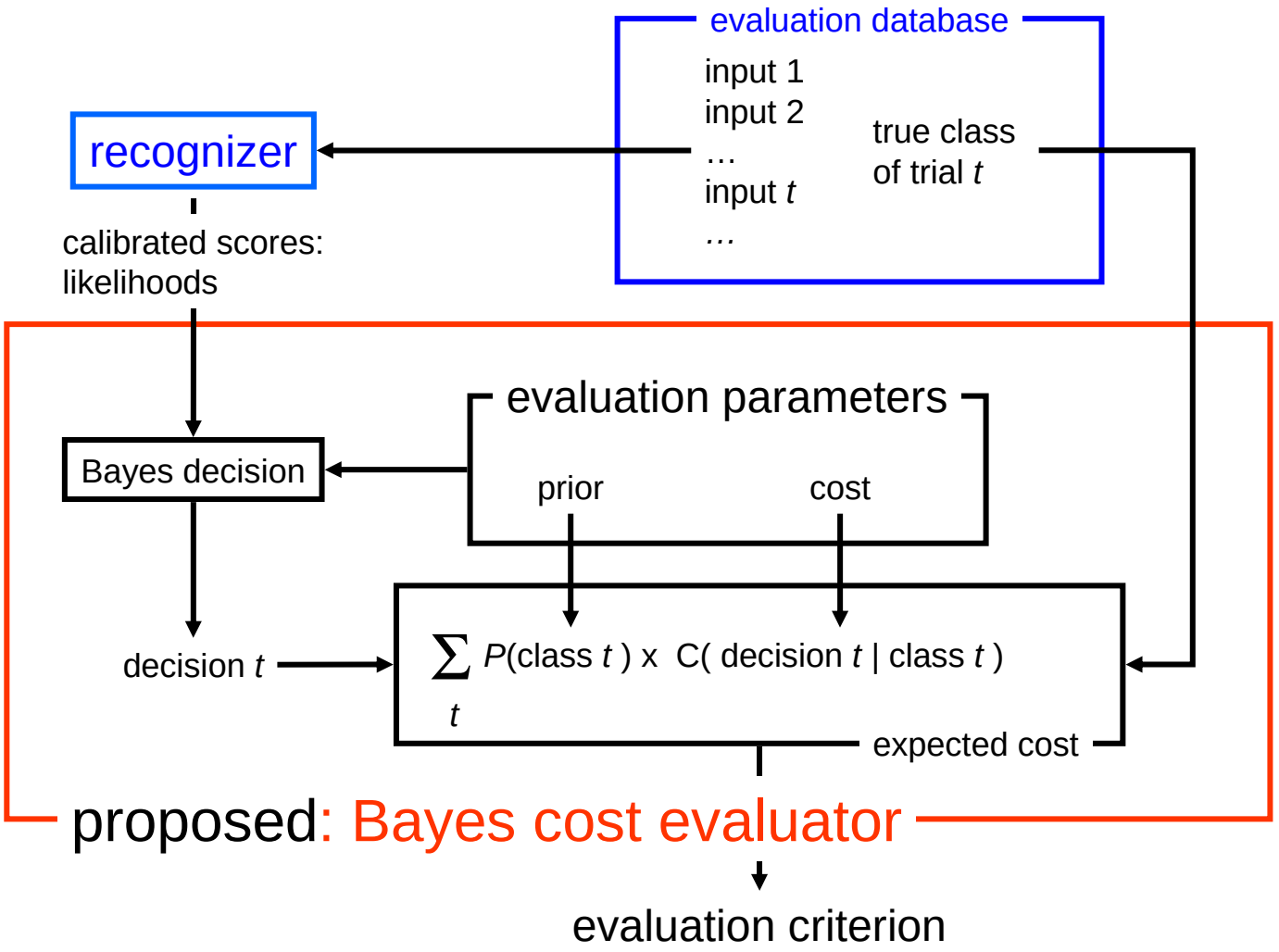
How does the evaluator judge the goodness of the recognizer's likelihoods?

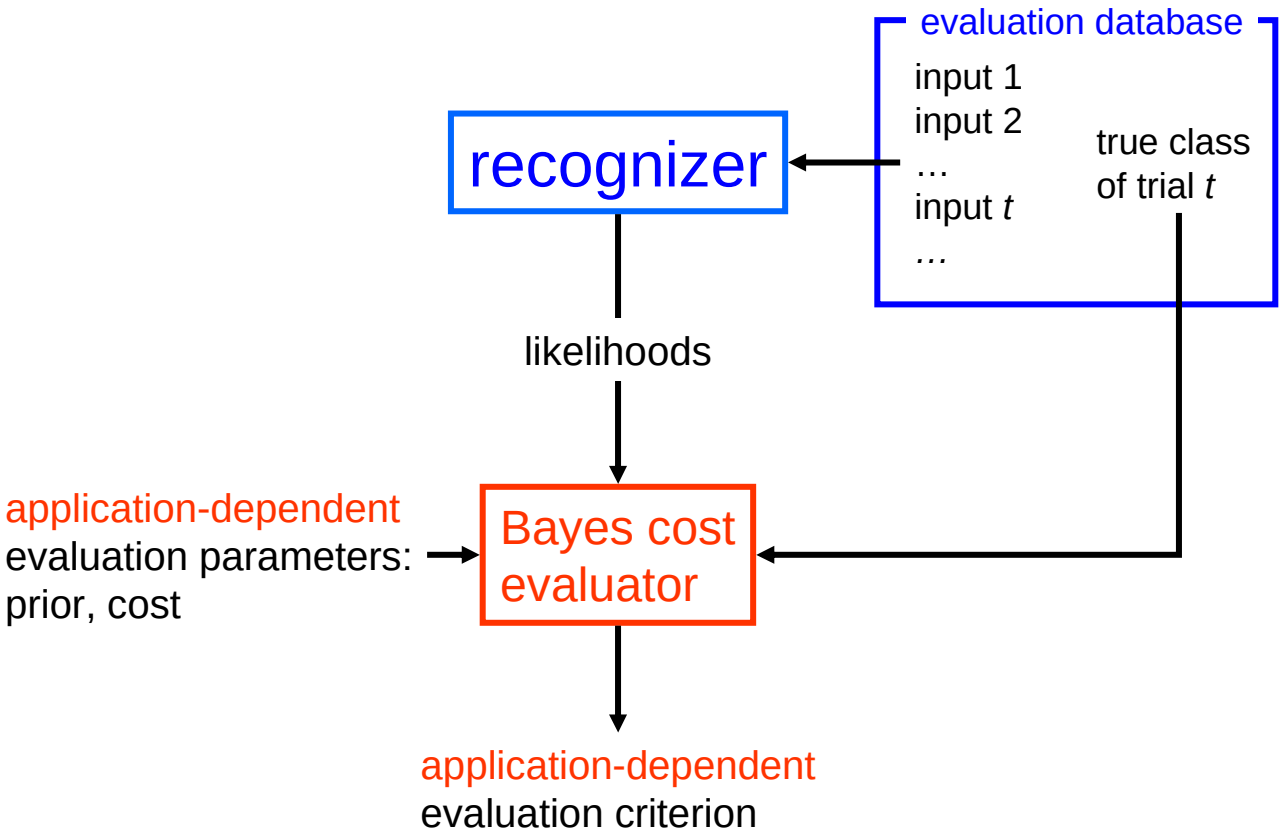
- Apply the likelihoods for their designed purpose: use them to make Bayes decisions.
- Evaluate the goodness (cost) of those decisions.

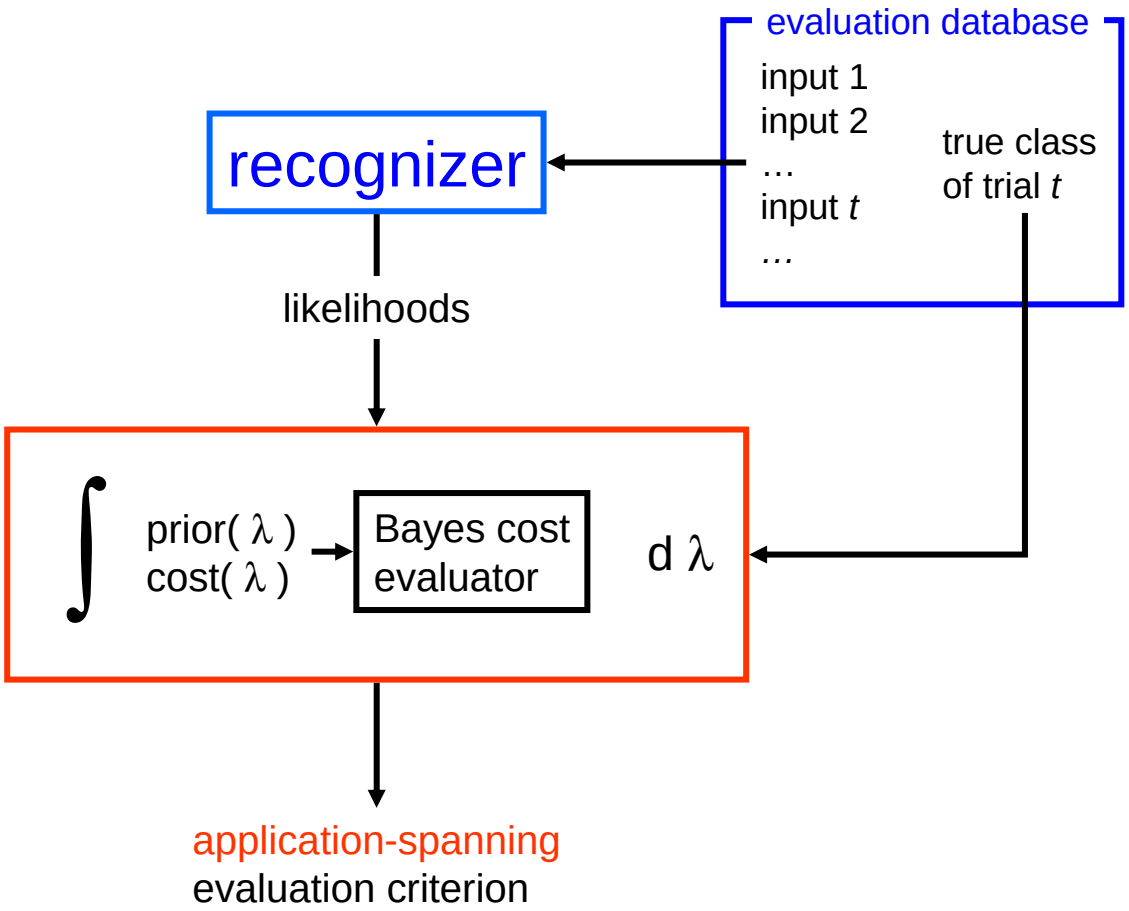

evaluation database
input 1
input 2
…
input *t*
…
true class
of trial *t*
black box
recognizer
?
decision *t*
fixed
evaluation parameters
prior
cost
$\sum_t$ *P*(class *t*) x C(decision *t* | class *t*)
expected cost
existing NIST recipe
evaluation criterion


evaluation database
input 1
input 2
…
input *t*
…
true class
of trial *t*
recognizer
uncalibrated
scores
variety of
ad hoc
solutions
decision *t*
fixed
evaluation parameters
prior
cost
$\sum_t$ *P*(class *t*) x C(decision *t* | class *t*)
expected cost
existing NIST recipe
evaluation criterion

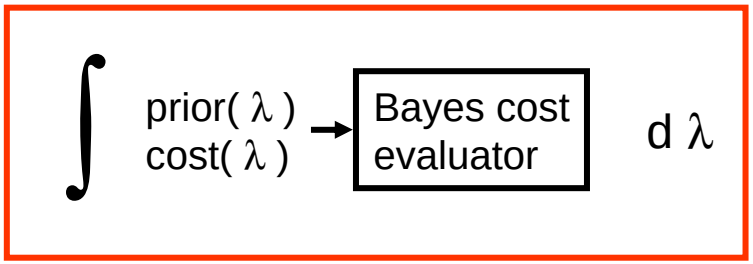

evaluation database
input 1
input 2
…
input t
…
true class
of trial t
recognizer
calibrated scores:
likelihoods
variable
evaluation parameters
Bayes decision
prior
cost
decision t
∑ P(class t) x C(decision t | class t)
t
expected cost
proposed evaluator
evaluation criterion


evaluation database
input 1
input 2
…
input t
…
true class
of trial t
recognizer
calibrated scores:
likelihoods
evaluation parameters
Bayes decision
prior
cost
decision t
∑ t P(class t) x C(decision t | class t)
expected cost
proposed: Bayes cost evaluator
evaluation criterion


evaluation database
input 1
input 2
…
input *t*
…
true class
of trial *t*
recognizer
likelihoods
application-dependent
evaluation parameters:
prior, cost
Bayes cost
evaluator
application-dependent
evaluation criterion


evaluation database
input 1
input 2
…
input *t*
…
true class
of trial *t*
recognizer
likelihoods
∫
prior(λ)
cost(λ)
Bayes cost
evaluator
d λ
application-spanning
evaluation criterion

Solving the integral



- With appropriate choices of the parametrizations, prior(λ) and cost(λ), this integral can be solved analytically.
- This solution forms a family of strictly proper scoring rules, a tool from statistics literature for the evaluation of the goodness of for example probabilistic weather forecasts.

We can now rewrite the integral (for the case $i = N$) as:

$$
\begin{aligned}
&\int_{\mathbb{P}_N} \Gamma(N) C^*_{\boldsymbol{\eta}}(\mathbf{p}|\theta_N)\, \mathbf{d}\boldsymbol{\eta} \\
&= \int_{-\infty}^{y_1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathcal{I}(\mathbf{x})\, dx_{N-1}\, dx_{N-2} \cdots\, dx_1 \\
&+ \int_{y_1}^{\infty} \int_{-\infty}^{y_2} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathcal{I}(\mathbf{x})\, dx_{N-1}\, dx_{N-2} \cdots\, dx_1 \\
&+ \int_{y_1}^{\infty} \int_{y_2}^{\infty} \int_{-\infty}^{y_3} \cdots \int_{-\infty}^{\infty} \mathcal{I}(\mathbf{x})\, dx_{N-1}\, dx_{N-2} \cdots\, dx_1 \\
&+ \cdots \\
&+ \int_{y_1}^{\infty} \int_{y_2}^{\infty} \int_{y_3}^{\infty} \cdots \int_{-\infty}^{y_{N-1}} \mathcal{I}(\mathbf{x})\, dx_{N-1}\, dx_{N-2} \cdots\, dx_1
\end{aligned}
$$

where

$$
\mathcal{I}(\mathbf{x}) = \frac{\Gamma(N-1)}{\eta_N} \prod_{k=1}^{N} \eta_k = \Gamma(N-1) \frac{e^{\sum_{k=1}^{N-1} x_k}}{\left(1 + \sum_{k=1}^{N-1} e^{x_k}\right)^{N-1}}
$$

Bayes cost evaluation
Summary

The recognizer's likelihoods are evaluated by how good the decisions are that those likelihoods can make. This gives a family of practical evaluation recipes, which can be used in two ways:

1. Evaluation over a range of different application parameters (prior, cost), which can be displayed graphically.
2. Or integrated, to provide an application-spanning, scalar measure of goodness.

Theory
Calibration

Calibration

Example of a calibration problem

Tue	Wed	Thu	Fri	Sat
68	72	64	63	64

Predicted[1] maximum temperature for the next 5 days. Can this be an accurate prediction?

[1] http://www.weathersa.co.za

Calibration

Example of a calibration problem

	Tue	Wed	Thu	Fri	Sat
°F	68	72	64	63	64
$°C = \frac{5}{9}(°F - 32)$	20	22	18	17	18

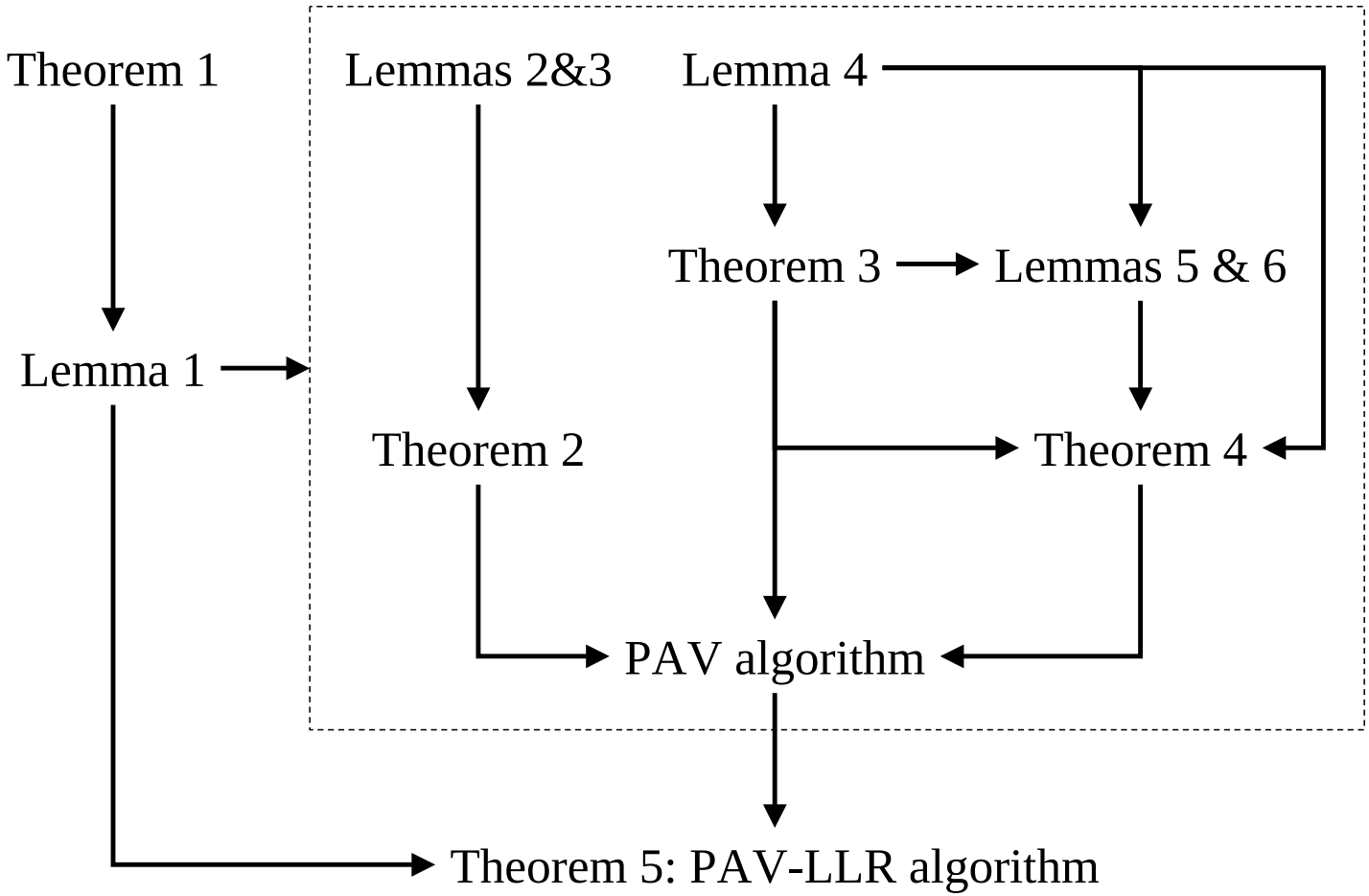
Predicted maximum temperature for the next 5 days, now re-calibrated, so that we can understand it.

Calibration

Probabilities and likelihoods can also be badly calibrated in the same way as the above temperature prediction: all the information is there, but not in the format we expect it. In this work we propose ways to:

- Measure the degree of miscalibration of a pattern recognizer.
- Re-calibrate the recognizer to improve calibration.

All of this is still based on Bayes decisions.


Theorem 1
Lemmas 2&3
Lemma 4
Theorem 3
Lemmas 5 & 6
Lemma 1
Theorem 2
Theorem 4
PAV algorithm
Theorem 5: PAV-LLR algorithm

Theory
Discriminative training

Discriminative training
FoCal Toolkit

A scalar evaluation criterion (such as we formed with the above integral) is the most important ingredient needed for discriminative training.

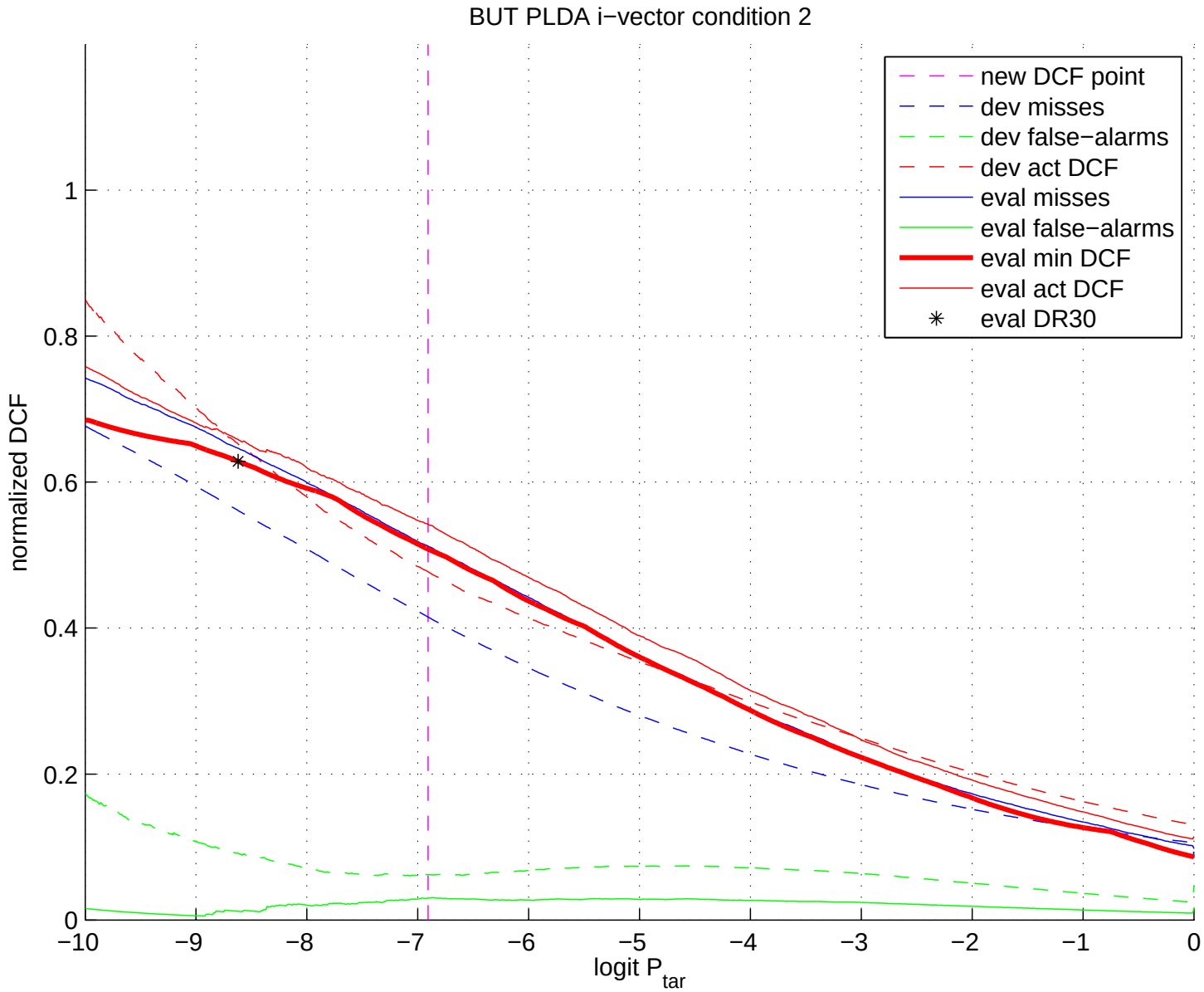
- In particular, one of the members of this family, the logarithmic proper scoring rule, has many desirable properties as a discriminative training objective function.
- The FoCal Toolkit (the practical embodiment of this work, which is used by many other researchers) uses the logarithmic scoring rule as discriminative training criterion to optimize calibration and fusion transformations of the scores of speaker and language recognizers.

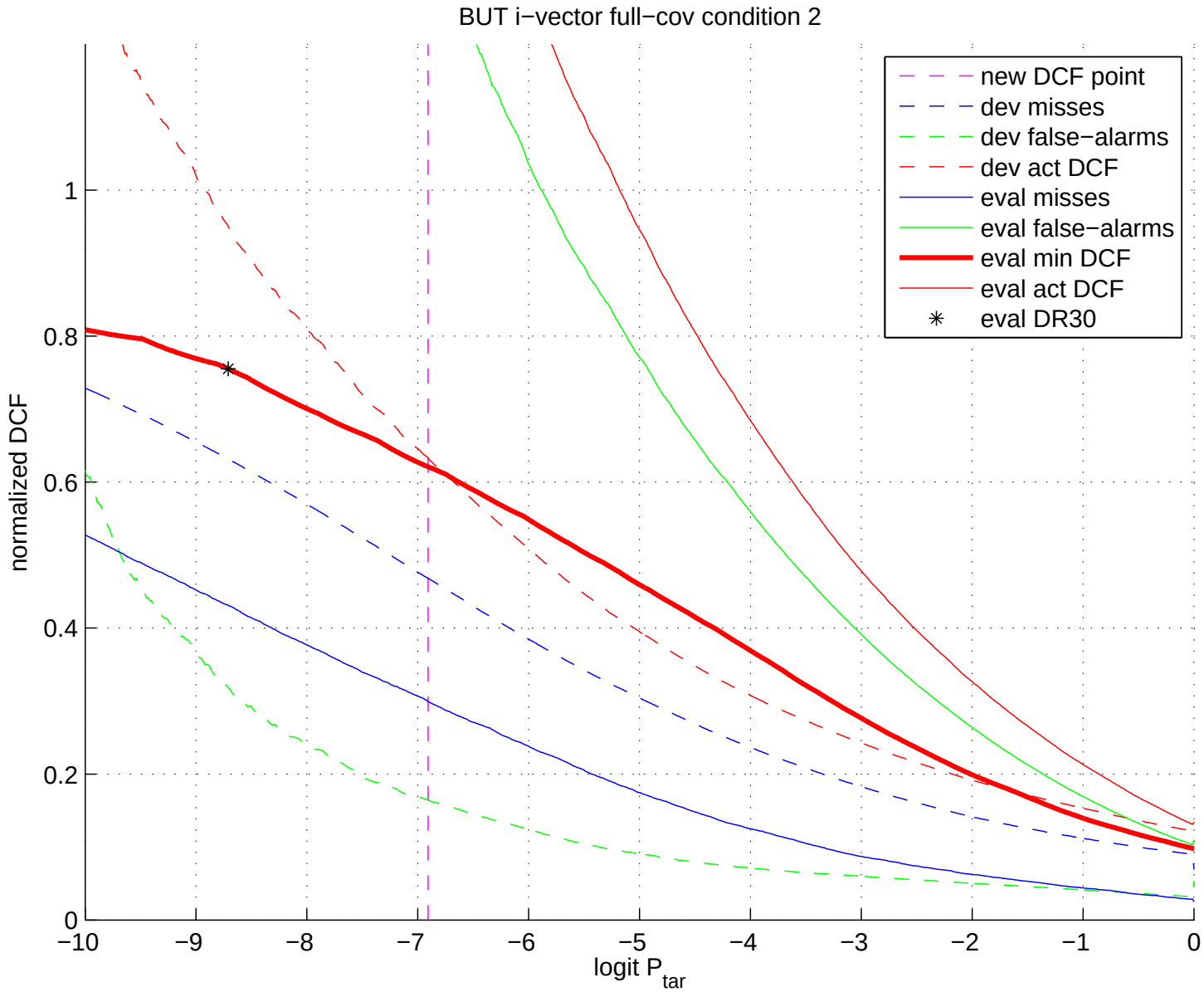
Examples

Speaker Recognition Examples
Calibration analysis

Below we show two examples of our evaluation methods as applied to two different speaker recognition systems in the NIST 2010 Speaker Recognition Evaluation.

1. The first example shows a recognizer with good calibration over a wide range of different operating points.
2. The second example shows a recognizer which could have been good, but bad calibration spoils the applicability of this recognizer at most operating points.

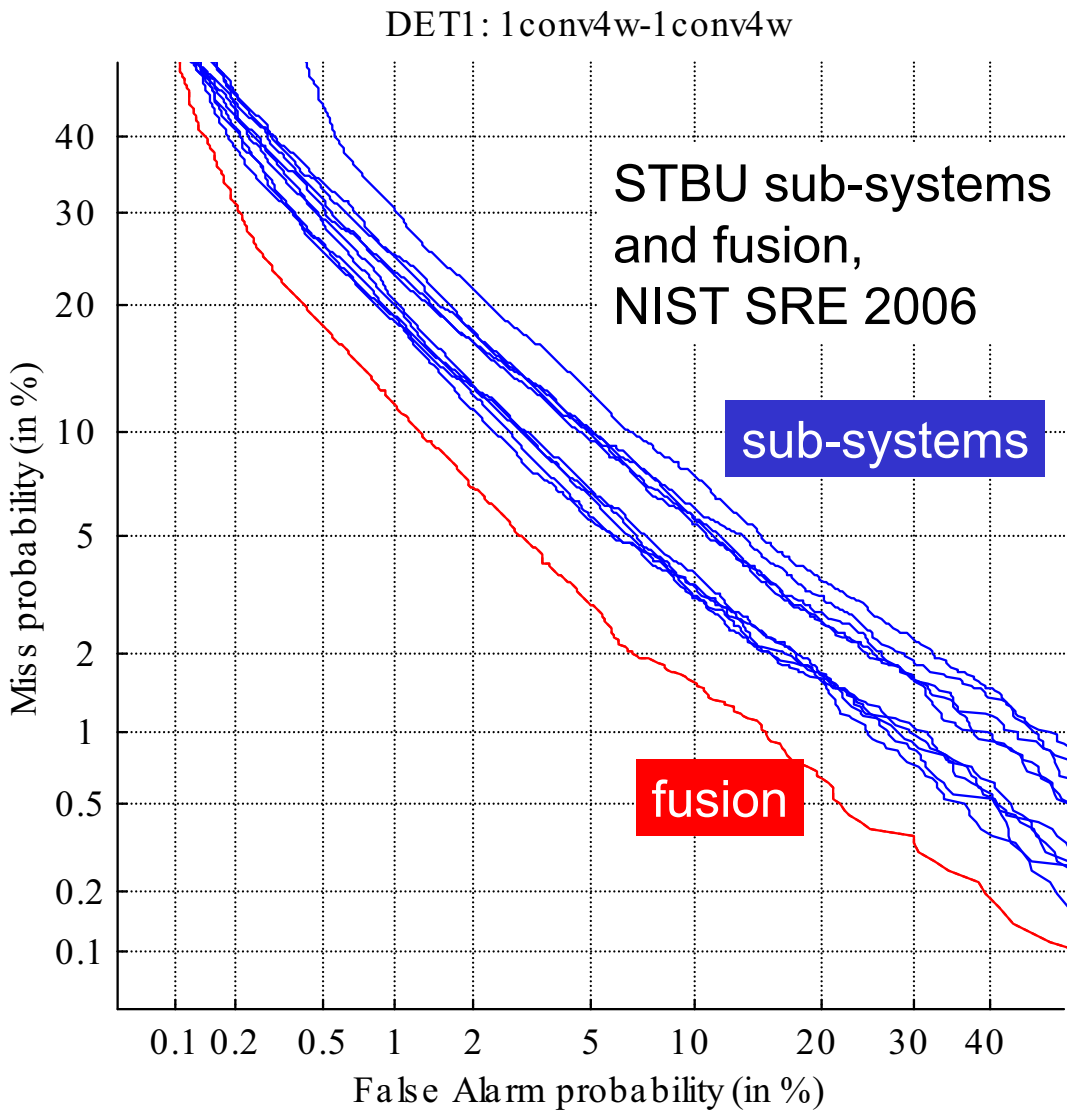

BUT PLDA i−vector condition 2
new DCF point
dev misses
dev false−alarms
dev act DCF
eval misses
eval false−alarms
eval min DCF
eval act DCF
eval DR30
normalized DCF
logit P tar
1
0.8
0.6
0.4
0.2
0
−10
−9
−8
−7
−6
−5
−4
−3
−2
−1
0


BUT i−vector full−cov condition 2
new DCF point
dev misses
dev false−alarms
dev act DCF
eval misses
eval false−alarms
eval min DCF
eval act DCF
eval DR30
normalized DCF
logit P tar
0
0.2
0.4
0.6
0.8
1
−10
−9
−8
−7
−6
−5
−4
−3
−2
−1
0

Speaker Recognition Example

Discriminatively trained fusion

Below is an example of a discriminatively trained fusion of multiple speaker recognition subsystems in the NIST 2006 Speaker Recognition Evaluation. The fusion shows a dramatic improvement in accuracy.

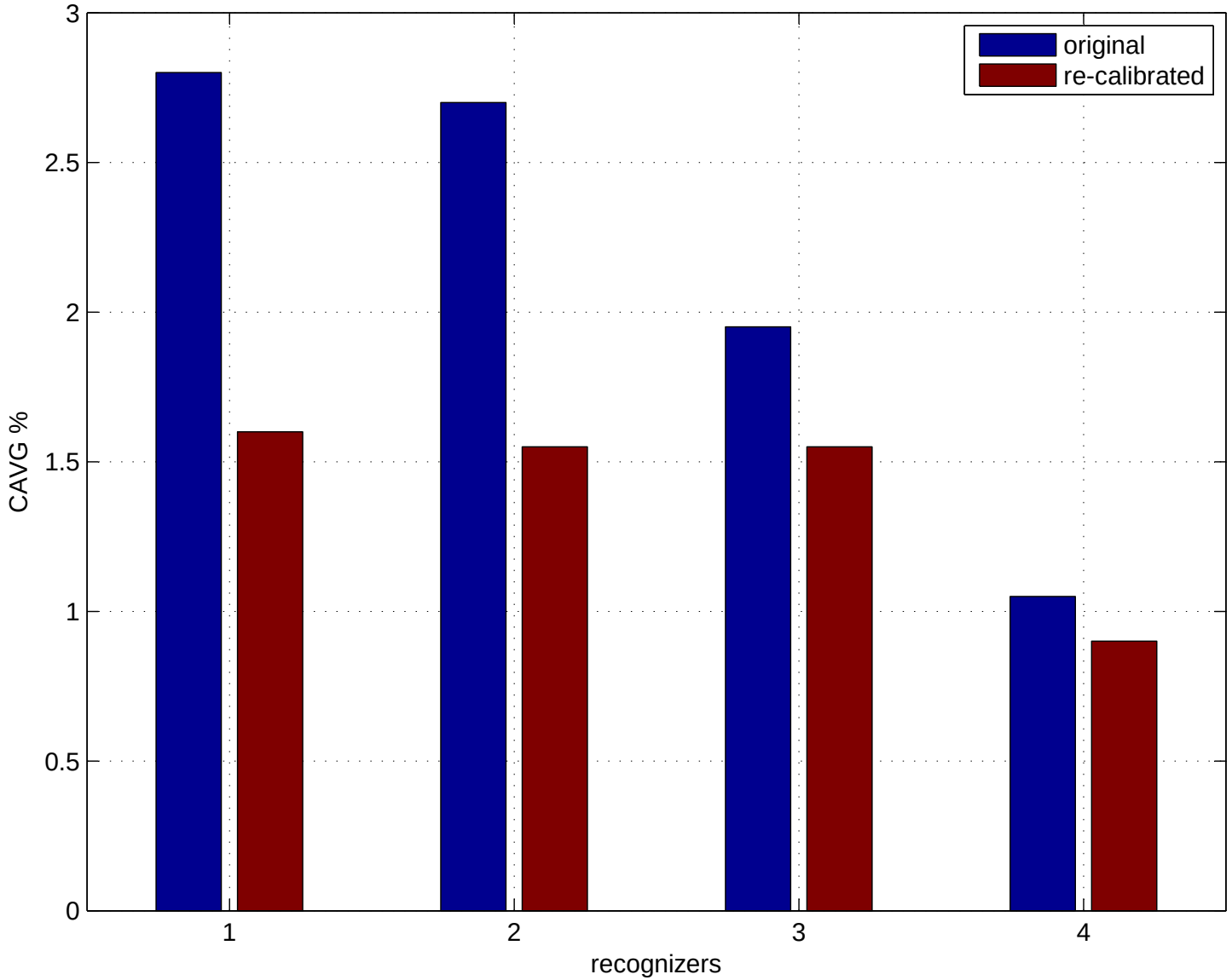

DET1: 1conv4w-1conv4w
STBU sub-systems and fusion, NIST SRE 2006
sub-systems
fusion
Miss probability (in %)
False Alarm probability (in %)
40
30
20
10
5
2
1
0.5
0.2
0.1
0.1 0.2 0.5 1 2 5 10 20 30 40

Language Recognition Example

Discriminatively trained calibration

The next slide shows an example of four different language recognizers, submitted by other researchers to the NIST 2007 Language Recognition Evaluations.

- The blue bars show the error-rates of the originally submitted systems.
- The red bars show the improved accuracy of the the same systems, re-calibrated by using the FoCal Toolkit.


original
re-calibrated
CAVG %
recognizers
0
0.5
1
1.5
2
2.5
3
1
2
3
4

Conclusion

Conclusion

In summary:

- Pattern recognizers with likelihood outputs are good for making Bayes decisions.
- Bayes decisions are good for evaluating such pattern recognizers.
- Such evaluation is good for building better recognizers.

The dissertation and associated software can be downloaded from `http://niko.brummer.googlepages.com`.