Tractable priors, likelihoods, posteriors and proper scoring rules for the astronomically complex problem of partitioning a large set of recordings w.r.t. speaker

Niko Brümmer

Nuance Communications, South Africa

VGS Invited Talks @ FIT
Brno University of Technology
April 2018

Outline

Outline

Motivation: The familiar binary case

The canonical binary problem

Given $\mathcal{R}$, a set of 2 recordings, were they spoken by the same speaker (H_1), or by two different speakers (H_2)?

Tools for probabilistic solutions

Prior: $\pi = P(H_1 \mid \pi) = 1 - P(H_2 \mid \pi)$

Likelihood: $\lambda = \frac{P(\mathcal{R}|H_1,\mathcal{M})}{P(\mathcal{R}|H_2,\mathcal{M})}$ for some speaker recognizer, $\mathcal{M}$

Posterior: $P(H_1 \mid \lambda, \pi) = \frac{\pi\lambda}{\pi\lambda+1-\pi} = 1 - P(H_2 \mid \lambda, \pi)$

Bayes dec.: $\hat{D} = \text{argmin}_{D\in\mathcal{D}} \left\langle C(D,h) \right\rangle_{P(h|\lambda,\pi)}$

and for judging the goodness of those solutions

Proper scoring rule: $S(\lambda; H_{\text{true}}) = -\log P(H_{\text{true}} \mid \lambda, \pi)$

Motivation: The real world

In the real world, speaker recognition problems do not occur only in the form of neat, NIST-style binary verification trials and we are not always provided with convenient, labelled training databases.

In the most general cases, we are merely given a (possibly large) unsupervised set of recordings.

- It would be really useful if we can just go and recognize the speakers in there.
- This problem is known as speaker clustering or speaker partitioning.

Motivation: The real world

The complication is that (just like in the binary case), speaker clustering cannot find the correct solution with certainty and a principled answer to this problem has to be probabilistic.

> *Unfortunately, a probabilistic treatment of clustering is a lot harder than in the binary case.*

We propose some solutions in this talk.

Outline

Dramatis personae

$\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$: a set of n speech recordings. We assume each recording contains one speaker.

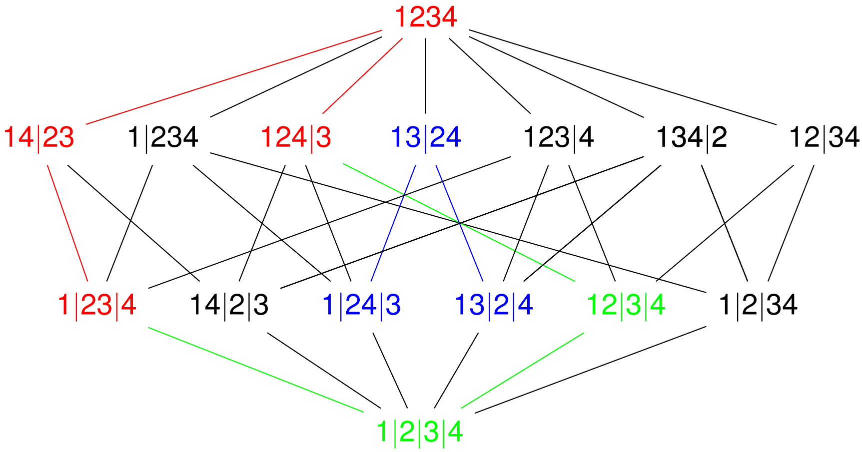
$\mathcal{P}_n$: The set of all possible partitions of the index set, $\mathcal{I}_n = \{1, 2, \ldots, n\}$.

B_n: The Bell number. The size of $\mathcal{P}_n$.

n	B_n	$\mathcal{P}_n$
1	1	$\{1\}$
2	2	$\{12, 1\|2\}$
3	5	$\{123, 1\|23, 2\|13, 3\|12, 1\|2\|3\}$
4	15	see next slide $\Longrightarrow$
$\ldots$		
70	10^{80}	$\approx$ number of atoms in the known universe

Lattice structure of $\mathcal{P}_n$

Hasse diagram for $n = 4$



Dramatis personae

Partition likelihoods

$\mathcal{L}, \mathcal{L}' \in \mathcal{P}_n$. Partitions of $\mathcal{R}$—sets of speaker labels for the entire set of n recordings.

$\mathcal{M}$: The parameters of a probabilistic speaker recognition model.

Generative model

Computes partition likelihoods:

$$P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}), \text{for any } \mathcal{L} \in \mathcal{P}_n$$

Discriminative model

Computes partition likelihood-ratios of the form:

$$\mathrm{LR} = \frac{P(\mathcal{R} \mid \mathcal{L}, \mathcal{M})}{P(\mathcal{R} \mid \mathcal{L}', \mathcal{M})} = \frac{P(\mathcal{L} \mid \mathcal{R}, \mathcal{M})}{P(\mathcal{L}' \mid \mathcal{R}, \mathcal{M})} \frac{P(\mathcal{L}')}{P(\mathcal{L})}$$

Partition likelihoods

Partition likelihoods and likelihood-ratios are tractable—provided you don't have to compute all B_n of them!

Generative models

- Gaussian PLDA, Two-covariance model (since Odyssey'10)
- Heavy-tailed PLDA (Odyssey'10 and '18, Interspeech'18)
- Deep generative models (future ...)

Discriminative models

We are currently working on these:
`github.com/bsxfan/meta-embeddings`

Some properties of the PLDA model

Maximum likelihood training is tractable (even fast)

The partition likelihood is familiar from the EM algorithm for PLDA training. Given a supervised database, $\mathcal{R}$, with true partition, $\mathcal{L}^*$, the EM-algorithm finds the maximum likelihood parameter estimate:

$$\hat{\mathcal{M}} = \operatorname*{argmax}_{\mathcal{M}} P(\mathcal{R} \mid \mathcal{L}^*, \mathcal{M})$$

…optimal clustering is not

For large n, finding the exact maximum likelihood partition:

$$\hat{\mathcal{L}} = \operatorname*{argmax}_{\mathcal{L} \in \mathcal{P}_n} P(\mathcal{R} \mid \mathcal{L}, \mathcal{M})$$

is (as far as we know) hopelessly intractable.

The intractable partition posterior

Recall: maximum likelihood (ML) training is tractable

$$\hat{\mathcal{M}} = \operatorname*{argmax}_{\mathcal{M}} P(\mathcal{R} \mid \mathcal{L}^*, \mathcal{M})$$

. . . maximum conditional likelihood (MCL) is not

$$\begin{aligned}\hat{\mathcal{M}} &= \operatorname*{argmax}_{\mathcal{M}} P(\mathcal{L}^* \mid \mathcal{R}, \mathcal{M}) \qquad \text{(partition posterior)} \\ &= \operatorname*{argmax}_{\mathcal{M}} \frac{P(\mathcal{R} \mid \mathcal{L}^*, \mathcal{M}) P(\mathcal{L}^*)}{\sum_{\mathcal{L} \in \mathcal{P}_n} P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}) P(\mathcal{L})}\end{aligned}$$

The intractable partition posterior

posterior via likelihoods (intractable)

$$P(\mathcal{L}^* \mid \mathcal{R}, \mathcal{M}) = \frac{P(\mathcal{R} \mid \mathcal{L}^*, \mathcal{M})P(\mathcal{L}^*)}{\sum_{\mathcal{L}\in\mathcal{P}_n} P(\mathcal{R} \mid \mathcal{L}, \mathcal{M})P(\mathcal{L})}$$

posterior via likelihood-ratios (intractable)

$$P(\mathcal{L}^* \mid \mathcal{R}, \mathcal{M}) = \frac{\frac{P(\mathcal{R}|\mathcal{L}^*,\mathcal{M})}{P(\mathcal{R}|\mathcal{L}',\mathcal{M})}P(\mathcal{L}^*)}{\sum_{\mathcal{L}\in\mathcal{P}_n} \frac{P(\mathcal{R}|\mathcal{L},\mathcal{M})}{P(\mathcal{R}|\mathcal{L}',\mathcal{M})}P(\mathcal{L})}$$

Bayes decisions

Often, full knowledge of the partition, $\mathcal{L}$ is not of ultimate interest. Instead, some decision, $D \in \mathcal{D}$, might be desired, where $\mathcal{D}$ is some simpler space. For example:

- $D =$ number of speakers in $\mathcal{R}$
- $D =$ the subset of speakers with more than 20 recordings each

We need a cost function, $C(D, \mathcal{L}) \to \mathbb{R}$, the cost of decision D, when $\mathcal{L}$ is the true partition. The minimum-expected-cost Bayes decision is:

$$\hat{D} = \underset{D \in \mathcal{D}}{\operatorname{argmin}} \sum_{\mathcal{L} \in \mathcal{P}_n} P(\mathcal{L} \mid \mathcal{R}, \mathcal{M}) C(D, \mathcal{L})$$

An exact computation is intractable, because of the summation.

The logarithmic proper scoring rule

For similar reasons to the binary case, it would be ideal if we could do the following:

- Given:
  - a model, $\mathcal{M}$
  - a supervised evaluation database: $\mathcal{R}, \mathcal{L}^*$
- How good is the model at computing the probabilistic clustering solution, $P(\mathcal{L} \mid \mathcal{R}, \mathcal{M})$?

A nice answer would be to compute the cost function (log scoring rule):

$$S(\mathcal{M}; \mathcal{R}, \mathcal{L}^*) = -\log P(\mathcal{L}^*, \mid \mathcal{R}, \mathcal{M})$$

But, as we already know, the posterior is intractable.

Unsupervised training

Given just $\mathcal{R}$, with no speaker labels, exact maximum likelihood unsupervised training:

$$\begin{aligned}\mathcal{M} &= \underset{\mathcal{M}'}{\operatorname{argmax}} \, P(\mathcal{R} \mid \mathcal{M}) \\ &= \underset{\mathcal{M}'}{\operatorname{argmax}} \sum_{\mathcal{L}\in\mathcal{P}_n} P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}) P(\mathcal{L})\end{aligned}$$

is also hopelessly intractable.

Summary: Probabilistic Partitioning

tractable (for suitable models)

- everything with small n
- supervised ML training (large n)

intractable for large n (exact solutions)

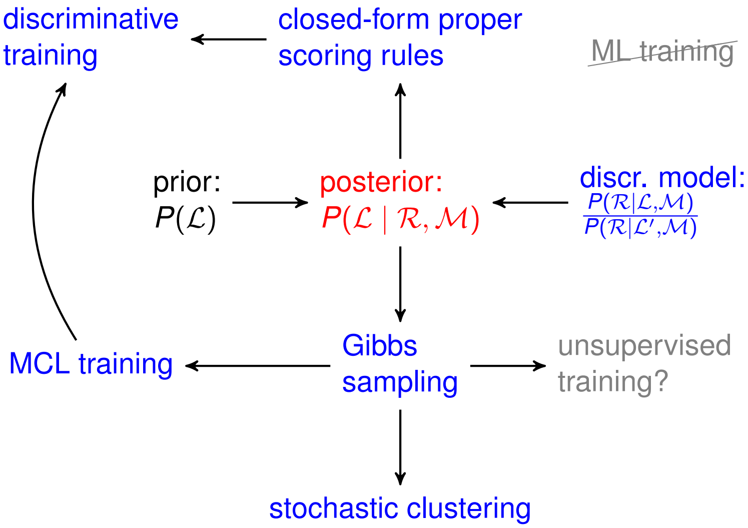
- supervised MCL training
- log scoring rule,
- Bayes decisions via posterior expectation
- clustering: ML, most probable
- unsupervised ML training

Outline

Tractable solutions (generative models)

$$\begin{tikzcd}
\text{discriminative training} & \text{closed-form proper scoring rules} \arrow[l] & \text{ML training} \\
\text{prior: } P(\mathcal{L}) \arrow[r] & \text{posterior: } P(\mathcal{L} \mid \mathcal{R}, \mathcal{M}) \arrow[u] \arrow[d] & \text{gen. model: } P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}) \arrow[l] \arrow[u] \\
\text{MCL training} \arrow[uu, bend left] & \text{Gibbs sampling} \arrow[l] \arrow[r] \arrow[d] & \text{unsupervised training} \\
 & \text{stochastic clustering} &
\end{tikzcd}$$

Tractable solutions (discriminative models)



The partition prior
Why do we need it?

Alert attendees may have noticed the appearance of an unannounced character, the partition prior, $P(\mathcal{L})$.

- Our speaker recognition models can give us likelihoods/likelihood-ratios, but to convert those to posteriors, we also need a prior.
- Even though the posterior is intractable, we shall need the posterior to be well defined—and for that we do need the prior.

How does one define probability distributions over spaces as large and complex as $\mathcal{P}_n$?

The CRP partition prior

The Chinese Restaurant Process (CRP) gives a convenient solution to define a prior. It is parametrized by two scalar parameters: α, β.

It allows computation of:

- $P(\mathcal{L} \mid \alpha, \beta)$, for any $\mathcal{L} \in \mathcal{P}_n$
- $P(\ell_i \mid \mathcal{L}_{\setminus i}, \alpha, \beta)$, for any $1 \leq i \leq n$
- The expected number of speakers given n, α, β.
- ML parameter estimate: $\text{argmax}_{\alpha,\beta} P(\mathcal{L}^* \mid \alpha, \beta)$

For more details, see:

- `en.wikipedia.org/wiki/Chinese_restaurant_process`
- Brümmer et al., "Meta-embeddings: A probabilistic generalization of embeddings in machine learning". `github.com/bsxfan/meta-embeddings`

Gibbs sampling the posterior

The problem

$$
\begin{aligned}
\mathcal{R} &= \{r_1, \ldots, r_n\}, \text{ set of recordings } (n \text{ large}) \\
\mathcal{L} &\in \mathcal{P}_n, \text{ a partition of } \mathcal{R} \text{ w.r.t. speaker} \\
B_n &: \text{the size of } \mathcal{P}_n \\
P(\mathcal{L} \mid \mathcal{R}) &: \text{partition posterior, intractable: } B_n \text{ too large}
\end{aligned}
$$

Divide and conquer

For any $1 \leq i \leq n$, decompose:

$$\mathcal{R} = (r_i, \mathcal{R}_{\setminus i}) \quad \text{and} \quad \mathcal{L} = (\ell_i, \mathcal{L}_{\setminus i})$$

where

$$
\begin{aligned}
\mathcal{R}_{\setminus i}, \mathcal{L}_{\setminus i} &: \text{obtained from } \mathcal{R}, \mathcal{L} \text{ by removing } r_i \\
\ell_i &: \text{speaker label for } r_i\text{, speakers hypothesized by } \mathcal{L}_{\setminus i}
\end{aligned}
$$

Given a generative/discriminative model that can compute partition likelihood-ratios for any $\mathcal{L}, \mathcal{L}' \in \mathcal{P}_n$, the conditional posterior:

$$P(\ell_i \mid \mathcal{L}_{\setminus i}, \mathcal{R}, \mathcal{M})$$

is tractable:

> *We can compute these probabilities and we can sample from them.*

The Gibbs sampling algorithm

"Stochastic clustering"

require: data, $\mathcal{R}$; model, $\mathcal{M}$; and CRP prior parameters, α, β

output: samples, $\mathcal{L} \in \mathcal{P}_n$, from $P(\mathcal{L} \mid \mathcal{R}, \mathcal{M})$

initialize: choose some $\mathcal{L} \in \mathcal{P}_n$, e.g. coarsest, or finest partition, or a sample from CRP prior

iterate:

- choose i (randomly or round-robin)
- decompose $\mathcal{L} \rightarrow \ell_i, \mathcal{L}_{\setminus i}$
- resample $\ell'_i \sim P(\ell_i \mid \mathcal{L}_{\setminus i}, \mathcal{R}, \mathcal{M})$
- reassemble $\mathcal{L} \leftarrow \ell'_i, \mathcal{L}_{\setminus i}$
- output the sample, $\mathcal{L}$

Stochastic clustering

Stochastic clustering is an alternative to AHC (agglomerative hierarchical clustering):

- Initialize with the finest partition (*n* speakers).
- Run the Gibbs sampler until it has 'warmed up'.
- Output any sample $\mathcal{L} \sim P(\mathcal{L} \mid \mathcal{R}, \mathcal{M})$.

This does not find the maximum likelihood partition, nor the most probable partition. But with high probability, we will find 'good' partitions with high posterior probability.

In cases where full detail of the partition, $\mathcal{L}$ is not required, but instead, some decision, $D \in \mathcal{D}$, approximate minimum-expected-cost Bayes decisions can be made as follows:

- Initialize with the finest partition (n speakers).
- Run the Gibbs sampler until it has 'warmed up'.
- Output:

$$\hat{D} = \underset{D \in \mathcal{D}}{\text{argmin}} \sum_{\mathcal{L} \sim P(\mathcal{L}|\mathcal{R},\mathcal{M})} C(D, \mathcal{L})$$

  where $C(D, \mathcal{L}) \to \mathbb{R}$ is the cost of decision D, when $\mathcal{L}$ is the true partition.

We experimented with this Gibbs sampler, applied to an i-vector PLDA recognizer:

- It worked for up to 10 or 20 speakers, but not for a typical large training database.
- The problem is that it changes but a single speaker label, ℓ_i, per iteration. There is not enough movement and the sampler has a high probability to remain stuck for long periods in some local, suboptimal mode of the posterior.

Can we fix this?

What can we do to improve the Gibbs sampler?

- The warm-up phase can be understood as a non-greedy, stochastic search for high probability partitions—it usually moves uphill, but not always.
- How can the sampler be modified to search more efficiently? Can we make it behave more like AHC?
- Can we generalize the decomposition $\mathcal{L} = (\ell_i, \mathcal{L}_{\setminus i})$? For example:

$$\mathcal{L} = (\text{one speaker}, \text{the rest})$$

- Can the posterior be temporarily smoothed with deterministic annealing?

More efficient Gibbs sampling

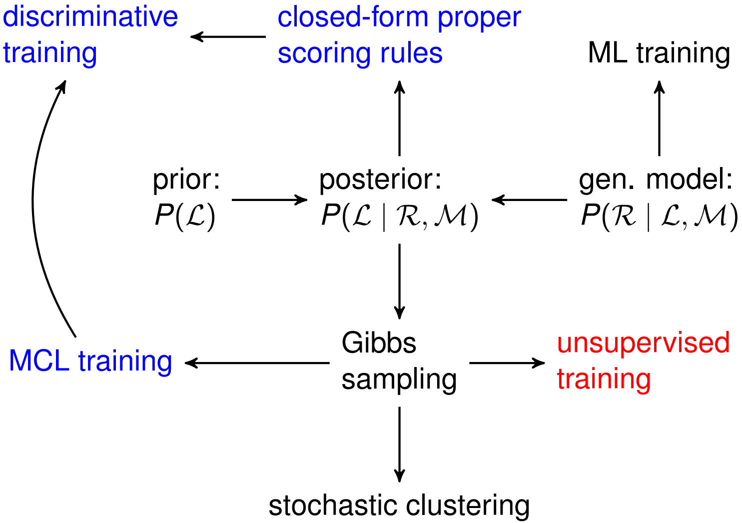
A proposed algorithm

Each iteration starts with some partition $\mathcal{L} \in \mathcal{P}_n$, which hypothesizes a number of 'speakers', say K of them. We alternate between two kinds of iteration:

agglomerate: Choose one 'test' speaker and compute the conditional posterior distribution for the 'open-set classification' problem having the other $K-1$ speakers for 'enrollment'. Sample from this posterior (one of K choices) and merge the test speaker into one of the other clusters, if needed.

split: Choose one speaker and sample from the posterior for the smaller partitioning problem for the recordings of this speaker. Split into multiple speakers if needed.

Unsupervised training



Unsupervised training
How not to do it

Let's consider:

$$\begin{aligned}\hat{\mathcal{M}} &= \operatorname*{argmax}_{\mathcal{M}} P(\mathcal{R} \mid \mathcal{M}) \\ &= \operatorname*{argmax}_{\mathcal{M}} \sum_{\mathcal{L}\in\mathcal{P}_n} P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}) P(\mathcal{L}) \\ &\approx \operatorname*{argmax}_{\mathcal{M}} \frac{1}{N} \sum_{\mathcal{L}\sim P(\mathcal{L})} P(\mathcal{R} \mid \mathcal{L}, \mathcal{M})\end{aligned}$$

Sampling from the dumb, clueless prior is a really bad idea! You will wait forever for it to accidentally hit the sharp maximum-likelihood peak $\max_{\mathcal{L}} P(\mathcal{R} \mid \mathcal{L}, \mathcal{M})$. An affordable number of samples (N of them) will give a really poor approximation to the full sum.

We can do a stochastic approximation to the EM algorithm, with $\mathcal{L}$ as hidden variable. The EM auxiliary can be approximated as:

$$
\begin{aligned}
Q(\mathcal{M}', \mathcal{M}) &= \Big\langle \log P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}') \Big\rangle_{P(\mathcal{L}|\mathcal{R},\mathcal{M})} \\
&\approx \frac{1}{N} \sum_{\mathcal{L} \sim P(\mathcal{L}|\mathcal{R},\mathcal{M})} \log P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}')
\end{aligned}
$$

Compared to prior sampling, two things are better:

- The log flattens the peak of $\log P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}')$
- The posterior samples are in an area with high log-likelihood.

Alternatively, the benefits of posterior sampling can be exploited in a more modern machine learning recipe by doing stochastic gradient ascent (SGD) on:

$$
\begin{aligned}
\nabla_{\mathcal{M}} P(\mathcal{R} \mid \mathcal{M}) &= \Big\langle \nabla_{\mathcal{M}} \log P(\mathcal{R} \mid \mathcal{L}, \mathcal{M}) \Big\rangle_{P(\mathcal{L}|\mathcal{R},\mathcal{M})} \\
&\approx \frac{1}{N} \sum_{\mathcal{L} \sim P(\mathcal{L}|\mathcal{R},\mathcal{M})} \nabla_{\mathcal{M}} \log P(\mathcal{R} \mid \mathcal{L}, \mathcal{M})
\end{aligned}
$$

This recipe is similar to Geoff Hinton's contrastive divergence for training RBMs.

MCL training

discriminative training

closed-form proper scoring rules

~~ML training~~

prior: $P(\mathcal{L})$

posterior: $P(\mathcal{L} \mid \mathcal{R}, \mathcal{M})$

discr. model: $\frac{P(\mathcal{R}|\mathcal{L},\mathcal{M})}{P(\mathcal{R}|\mathcal{L}',\mathcal{M})}$

MCL training

Gibbs sampling

unsupervised training

stochastic clustering

MCL: max. conditional likelihood training
(discriminative training using log scoring rule)

Recall: Given a supervised database, $\mathcal{R}, \mathcal{L}^*$, exact MCL does:

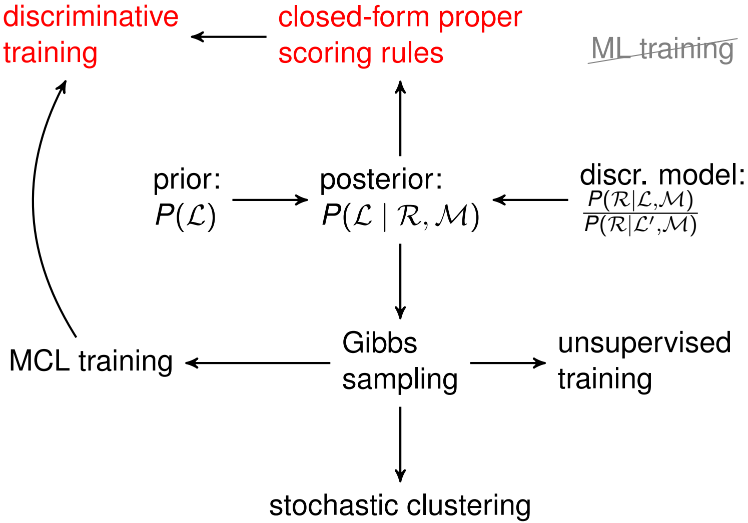
$$\hat{\mathcal{M}} = \underset{\mathcal{M}}{\operatorname{argmax}}\, P(\mathcal{L}^* \mid \mathcal{R}, \mathcal{M})$$

If we can sample from the posterior, we can approximate the gradient of this objective function as:

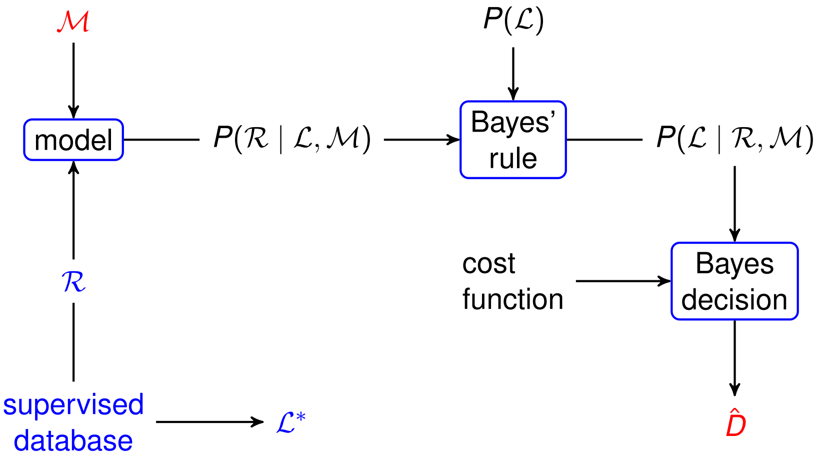
$$\begin{aligned}
&\nabla_{\mathcal{M}} \log P(\mathcal{L}^* \mid \mathcal{R}, \mathcal{M}) \\
&= \nabla_{\mathcal{M}} \log \tilde{P}(\mathcal{L}^* \mid \mathcal{R}, \mathcal{M}) - \Big\langle \nabla_{\mathcal{M}} \log \tilde{P}(\mathcal{L} \mid \mathcal{R}, \mathcal{M}) \Big\rangle_{P(\mathcal{L}|\mathcal{R},\mathcal{M})} \\
&\approx \nabla_{\mathcal{M}} \log \tilde{P}(\mathcal{L}^* \mid \mathcal{R}, \mathcal{M}) - \frac{1}{N} \sum_{\mathcal{L} \sim P(\mathcal{L}|\mathcal{R},\mathcal{M})} \nabla_{\mathcal{M}} \log \tilde{P}(\mathcal{L} \mid \mathcal{R}, \mathcal{M})
\end{aligned}$$

where $\tilde{P}(\mathcal{L} \mid \mathcal{R}, \mathcal{M})$ is a tractable (unnormalized) version of the posterior. This gives another contrastive-divergence-style optimization recipe.

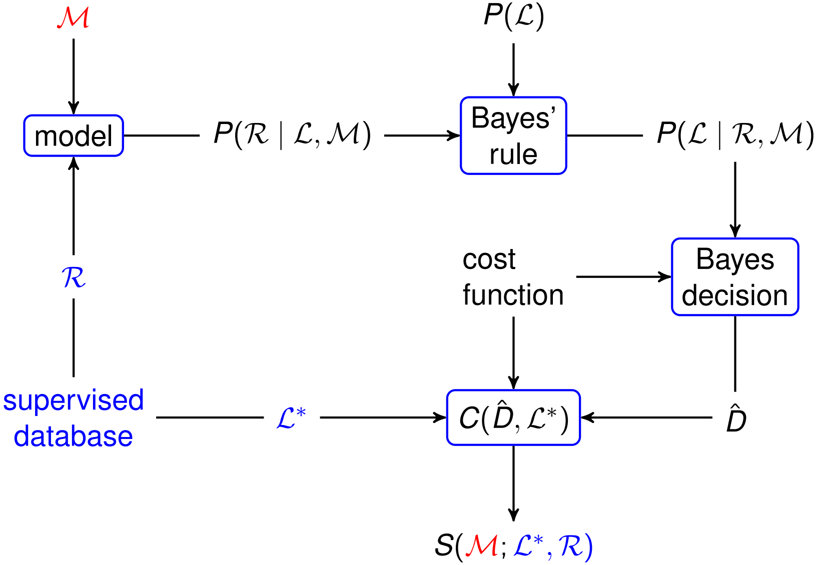
Closed-form proper scoring rules



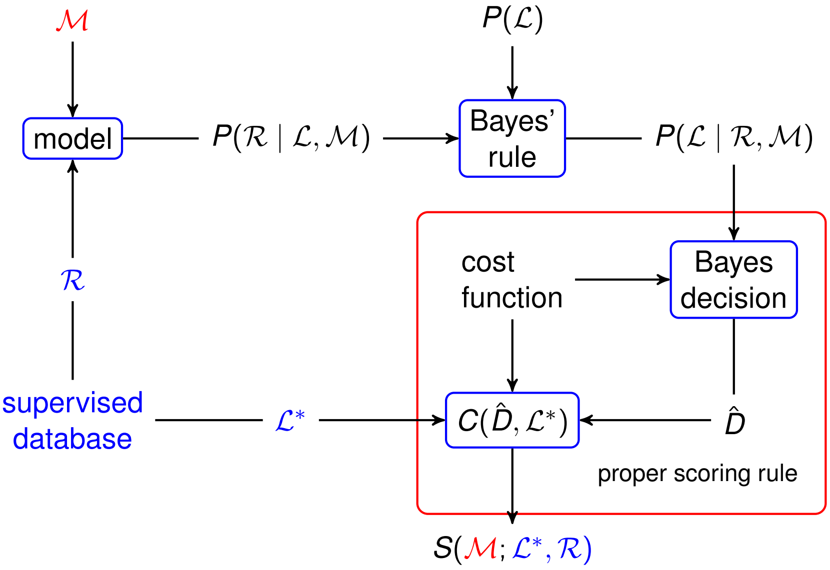
Proper scoring rule induced by Bayes decision



Proper scoring rule induced by Bayes decision



Proper scoring rule induced by Bayes decision



Proper scoring rule induced by Bayes decision

Advantages

- Represents cost of making a Bayes decision using the model
- Calibration-sensitive measure of goodness of the model
- Useful for evaluation
- Useful for discriminative training

Disadvantage?

The required expectation:

$$\sum_{\mathcal{L}\in\mathcal{P}_n} P(\mathcal{L} \mid \mathcal{R}, \mathcal{M}) C(D, \mathcal{L})$$

is intractable and cannot be computed exactly.

Closed-form proper scoring rules

- Although the expectation minimization:

$$\hat{D} = \underset{D \in \mathcal{D}}{\operatorname{argmin}} \sum_{\mathcal{L} \in \mathcal{P}_n} P(\mathcal{L} \mid \mathcal{R}, \mathcal{M}) C(D, \mathcal{L})$$

  is intractable,

- There do exist choices for the cost function, $C(D, \mathcal{L})$, such that the whole PSR:

$$S(\mathcal{M}; \mathcal{L}, \mathcal{R}) = C(\hat{D}, \mathcal{L})$$

  is tractable in exact, closed form.

Examples follow (without proof) $\Longrightarrow$

Closed-form proper scoring rules

Pseudolikelihood:

$$S(\mathcal{M};\mathcal{L}^*,\mathcal{R}) = -\sum_{i=1}^{n} \log P(\ell_i \mid \mathcal{L}_{\setminus i},\mathcal{R}), \qquad \forall i : \mathcal{L}^* = (\ell_i, \mathcal{L}_{\setminus i})$$

Composite likelihood:

$$S(\mathcal{M};\mathcal{L}^*,\mathcal{R}) = -\sum_{k} \log P(\mathcal{L}_k \mid \mathcal{L}'_k,\mathcal{R}), \qquad \forall k : \mathcal{L}_k, \mathcal{L}'_k \subset \mathcal{L}^*$$

$$\mathcal{L}_k \cap \mathcal{L}'_k = \emptyset$$

Composite likelihoods

Properties:

- composite likelihoods $\supset$ pseudolikelihood
- can be assembled from the same conditional posteriors as our Gibbs samplers
- they are proper scoring rules[1][2]
- closed form: no sampling needed
- can be used as discr. training criteria
- can be used as calibration-sensitive evaluation criteria of the goodness of probabilistic speaker recognition models

---

[1] Brümmer et al., "Meta-embeddings: A probabilistic generalization of embeddings in machine learning". `github.com/bsxfan/meta-embeddings`

[2] Dawid and Musio, "Theory and applications of proper scoring rules", `arxiv.org/abs/1401.0398`.

Summary
The last slide

- We have good tools for a Bayesian treatment of simple, binary speaker recognition trials
- In the real world, there are important, more general problems to be solved
- Combinatorial complexity complicates probabilistic solutions of these problems
- There are solutions to handle this complexity, some require sampling, some have closed forms.
- These solutions are new to speaker recognition and mostly untested. A lot of work remains to be done.