

Sampling Methods for Instantaneous Speaker Adaptation

CSTIT Project Presentation

Matt Shannon

Supervisor:
Mark Gales



1 July 2008

Outline

- 1 Introduction
 - Scenario
 - MLLR
 - Bayesian approach
 - Practical Bayesian adaptation
- 2 Sampling methods
 - Monte Carlo methods
 - High level Gibbs sampling
- 3 Results
 - Experimental set up
 - Results
- 4 Conclusion
 - Conclusions and future work

Scenario

Ultra low data adaptation:

- **single utterance** from a new, unseen speaker
- want to recognize that utterance as well as we can
- use **adaptation** to help us

Why not just use Maximum Likelihood Linear Regression (MLLR)?

MLLR

Linear mean-based adaptation:

- transform the mean of every Gaussian component by a **linear transform w** .

MLLR for recognition:

- somehow estimate the **maximum likelihood (ML)** transform
- **assume** this transform during recognition

Seems sensible enough!

Why not MLLR?

However:

- MLLR performs appallingly in single utterance case – worse than SI system!
- intuitively, we don't have enough data to get a good 'fix' on speaker's transform
- this is made clearer by considering **Bayesian perspective**

Bayesian approach to transforms

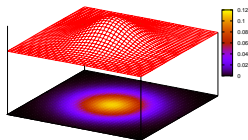
In Bayesian approach to adaptation:

- **transform w** just another random variable

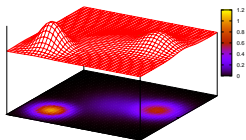
Bayesian approach to transforms

In Bayesian approach to adaptation:

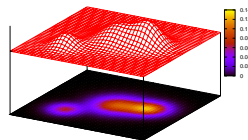
- transform w just another random variable



(d) prior $P(w)$



(e) likelihood $P(o|w)$



(f) posterior $P(w|o)$

$$\underbrace{P(w|o)}_{\text{posterior}} \propto \underbrace{P(o|w)}_{\text{likelihood}} \underbrace{P(w)}_{\text{prior}}$$

for transform w , and what we've heard o

Recognition as a weighted sum over transforms

Goal

Find word sequence h with highest probability $P(h|o)$ given data o

Recognition as a weighted sum over transforms

Goal

Find word sequence h with highest probability $P(h|o)$ given data o

For recognition with adaptation:

- have to consider **transform** w , since (fact from probability):

$$P(h|o) = \int P(h|w, o)P(w|o) dw$$

Recognition as a weighted sum over transforms

Goal

Find word sequence h with highest probability $P(h|o)$ given data o

For recognition with adaptation:

- have to consider **transform** w , since (fact from probability):

$$P(h|o) = \int P(h|w, o)P(w|o) dw$$

In other words, to get the **overall** belief $P(h|o)$ about the word sequence:

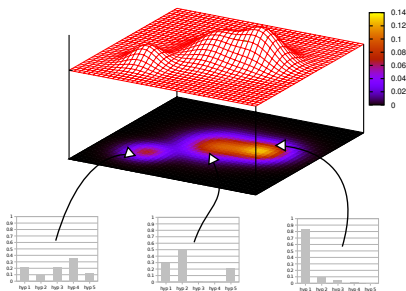
- we **average** our belief $P(h|w, o)$ about the word sequence **given each transform**
- **weighted** by the probability that transform is being used $P(w|o)$

Elegant!

Recognition as a weighted sum over transforms

$$P(h|o) = \int P(h|w, o)P(w|o) dw$$

Transform posterior $P(w|o)$ shown below. Each point is a transform.



Small graphs show distribution $P(h|w, o)$ over possible word sequences h

Why not MLLR? (again)

Now in a better position to see shortcomings of MLLR.

$$\text{Bayesian} \quad P(h|o) = \int P(h|w, o)P(w|o) dw$$

$$\text{MLLR} \quad P(h|o) \approx P(h|w_{\text{ML}}, o)$$

- MLLR **assumes** the ML transform w_{ML} when doing recognition, whereas Bayesian adaptation **marginalizes** over the transform w .
- fine if transform posterior $P(w|o)$ is heavily spiked at w_{ML}

Why not MLLR? (again)

In case of very little data:

- intuitively, haven't heard enough to get a 'fix' on speaker's transform
 - mathematically, transform posterior $P(w|o)$ is **diffuse**, meaning no big spikes
- ⇒ MLLR is liable to be a bad approximation

This is why we concentrate on case of very little data – should notice the biggest improvement of Bayesian over traditional methods.

Practical Bayesian adaptation

Bayesian approach

- elegantly incorporates all information we have about the transform w
 - but how to compute? Can't consider every transform!
 - indeed, turns out exact inference is pretty costly
- ⇒ we consider approximations

Practical Bayesian adaptation

Bayesian approach

- elegantly incorporates all information we have about the transform w
 - but how to compute? Can't consider every transform!
 - indeed, turns out exact inference is pretty costly
- ⇒ we consider approximations

Considered two classes of approximation in the project:

- **lower bound** schemes
- **sampling** schemes

Here we only talk about **sampling** schemes.

Outline

- 1 Introduction
 - Scenario
 - MLLR
 - Bayesian approach
 - Practical Bayesian adaptation
- 2 Sampling methods
 - Monte Carlo methods
 - High level Gibbs sampling
- 3 Results
 - Experimental set up
 - Results
- 4 Conclusion
 - Conclusions and future work

Monte Carlo integration

In general, can approximate the integral

$$\int f(x)p(x) dx$$

where f any function, p a distribution, by **Monte Carlo** approximation

$$\int f(x)p(x) dx \approx \frac{1}{n} \sum_i f(x_i)$$

where we take samples (x_i) from the distribution $p(x)$.

Monte Carlo integration

In general, can approximate the integral

$$\int f(x)p(x) dx$$

where f any function, p a distribution, by **Monte Carlo** approximation

$$\int f(x)p(x) dx \approx \frac{1}{n} \sum_i f(x_i)$$

where we take samples (x_i) from the distribution $p(x)$.

- great for low dimensional problems
 - in high dimensional case, f can be **very big** in just **a few** places
- ⇒ might never see these places where f big in a typical sampling run
- but they nevertheless contribute greatly to the true integral
- ⇒ have to be careful about how we choose f and p

Monte Carlo for adaptation – good method

Reminder – Goal

Find word sequence h with highest probability $P(h|o)$ given data o

Instead, let's try the expression we saw before:

$$P(h|o) = \int P(h|w, o)P(w|o) dw$$

Monte Carlo for adaptation – good method

Reminder – Goal

Find word sequence h with highest probability $P(h|o)$ given data o

Instead, let's try the expression we saw before:

$$P(h|o) = \int P(h|w, o)P(w|o) dw$$

- can compute $P(h|w, o)$ from $P(o|w, h)$, which we compute using Forwards-Backwards
 - sample from transform posterior $P(w|o)$ somehow!
 - $P(h|w, o)$ is bounded (probability), so never gets too high
- ⇒ good Monte Carlo method

Trick is how we sample from transform posterior $P(w|o)$!

High level Gibbs sampling

Skip derivation, but result is following 3-way iterative sampling procedure:

High level Gibbs sampling

- 1 sample the **word sequence** from $P(h|w, o)$ (consider all h)
 - 2 sample the **component sequence** from $P(\theta|h, o, w)$ (FB trick)
 - 3 sample the **transform** from $P(w|o, \theta)$ (Gaussian)
 - 4 (repeat)
- after sufficient iterations, generates a sample from $P(w|o)$
 - use this transform w as **one** of our sample points for Monte Carlo approximation

N -best rescoring framework

The first step was to sample from $P(h|w, o)$

- can compute this for any given word sequence h
 - but can't consider all possible word sequences
- ⇒ use an **N -best rescoring framework**:
- use speaker independent (SI) model to get top N word sequences
 - do Bayesian inference **assuming** word sequence is in this list
 - previous work on **Variational Bayes (VB)**, shows big gains over SI model even with $N = 5$
- ⇒ we used $N = 5$ for our experiments

Acoustic deweighting

- HMMs a terrible model – conditional independence assumptions really bad!
- ⇒ boost importance of language model (**language model scale factor**)
- ⇒ or reduce importance of acoustic model (**acoustic deweighting**)
 - but breaks **generative structure** of HMM formalism
 - turns out only **component level acoustic deweighting** is compatible with our sampling procedure
 - unfortunately component level deweighting seems to be pretty bad!

Outline

- 1 Introduction
 - Scenario
 - MLLR
 - Bayesian approach
 - Practical Bayesian adaptation
- 2 Sampling methods
 - Monte Carlo methods
 - High level Gibbs sampling
- 3 Results
 - Experimental set up
 - Results
- 4 Conclusion
 - Conclusions and future work

Experimental set up

- conversational telephone speech task
- single utterance
- test set has about 7000 utterances, about 3 seconds each

Results – component level deweighting

The performance of our **high level Gibbs sampling (HGS)** compared to the best known method **Variational Bayes (VB)** is shown below:

method	WER (%)
VB ¹	34.2
HGS	34.1

where we've used **component level acoustic deweighting of 0.01**.

¹actually something like VB – ask me

Results – component level deweighting

The performance of our **high level Gibbs sampling (HGS)** compared to the best known method **Variational Bayes (VB)** is shown below:

method	WER (%)
VB ¹	34.2
HGS	34.1

where we've used **component level acoustic deweighting of 0.01**.

- about the same performance as VB
- ⇒ approximations VB makes aren't too bad in this case

¹actually something like VB – ask me

Results – component level deweighting

The performance of our **high level Gibbs sampling (HGS)** compared to the best known method **Variational Bayes (VB)** is shown below:

method	WER (%)
VB ¹	34.2
HGS	34.1

where we've used **component level acoustic deweighting of 0.01**.

- about the same performance as VB

⇒ approximations VB makes aren't too bad in this case

However:

- there are signs transform posterior here is simpler than it should be

⇒ maybe this is why VB is a decent approximation in this case

¹actually something like VB – ask me

Results – overall

Amongst systems free to use any form of LM scale factor or acoustic deweighting:

method	WER (%)
SI	32.8
MLLR (150-best)	35.2
VB (5-best)	31.9
HGS	34.1

where all methods except HGS use a language model scale factor of 15.

Results – overall

Amongst systems free to use any form of LM scale factor or acoustic deweighting:

method	WER (%)
SI	32.8
MLLR (150-best)	35.2
VB (5-best)	31.9
HGS	34.1

where all methods except HGS use a language model scale factor of 15.

- not good!
 - yet performance given acoustic deweighting setting is very good
- ⇒ acoustic deweighting setting suboptimal
- but bad **value** or bad **type** of deweighting? Choice of type is restricted

Outline

- 1 Introduction
 - Scenario
 - MLLR
 - Bayesian approach
 - Practical Bayesian adaptation
- 2 Sampling methods
 - Monte Carlo methods
 - High level Gibbs sampling
- 3 Results
 - Experimental set up
 - Results
- 4 Conclusion
 - Conclusions and future work

Conclusions and future work

- developed and implemented **high level Gibbs sampling** procedure for linear mean-based adaptation that will **asymptotically** compute the **exact** posterior over an N -best list of word sequences

Conclusions and future work

- developed and implemented **high level Gibbs sampling** procedure for linear mean-based adaptation that will **asymptotically** compute the **exact** posterior over an N -best list of word sequences
- practical results inconclusive due to issues with acoustic deweighting
 - ⇒ investigate why component level deweighting works so badly
 - ⇒ is there some way to use a different type of acoustic deweighting with our high level Gibbs sampling?
 - ⇒ try other values of deweighting parameter

Conclusions and future work

- developed and implemented **high level Gibbs sampling** procedure for linear mean-based adaptation that will **asymptotically** compute the **exact** posterior over an N -best list of word sequences
- practical results inconclusive due to issues with acoustic deweighting
 - ⇒ investigate why component level deweighting works so badly
 - ⇒ is there some way to use a different type of acoustic deweighting with our high level Gibbs sampling?
 - ⇒ try other values of deweighting parameter
- all sorts of extensions possible, including:
 - single component Gibbs sampling with overrelaxation
 - using particle filter for component sequence sampling (may allow more general acoustic deweighting models)
 - using more general MCMC methods such as Hamiltonian Monte Carlo (would allow arbitrary transform priors)

References



David J. C. MacKay.

Information Theory, Inference, and Learning Algorithms.

Cambridge University Press, 2003.

Available from

<http://www.inference.phy.cam.ac.uk/mackay/itila/>.



K. Yu and M. J. F. Gales.

Bayesian adaptive inference and adaptive training.

IEEE Transactions on Audio, Speech and Language Processing,

15(6):1932–1943, 2007.