A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers

Conference on Empirical Methods in NLP, 2008

Jianfeng Gao
Microsoft Research

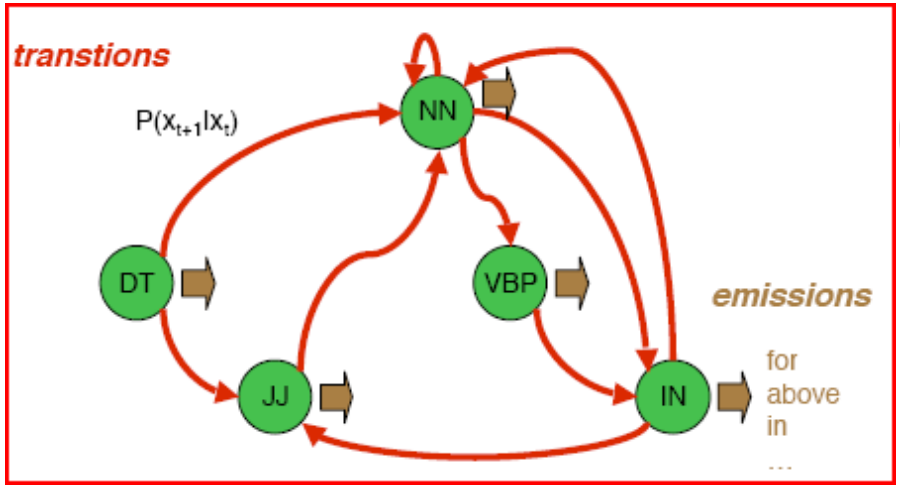
Mark Johnson
Brown Univeristy

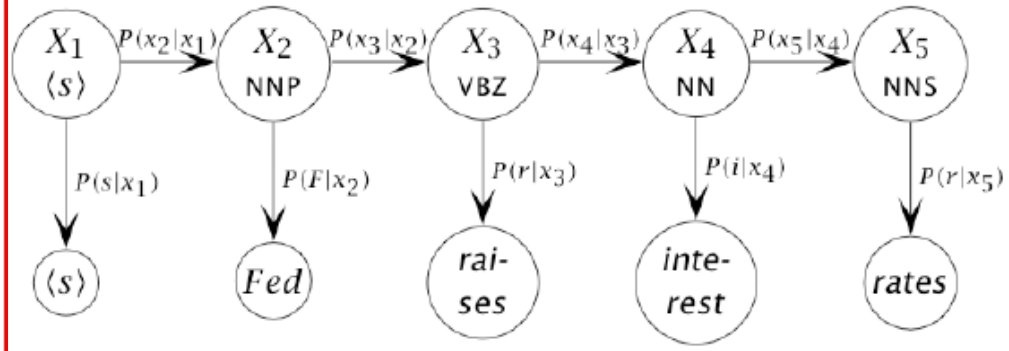
Presenter: Manish Gupta
Instructor: Dr. Julia Hockenmaier
CS598

24th Feb 2010

Basics

- Bayesian estimator: Estimator that minimizes posterior expected value of a loss function.
- Consider an unknown parameter θ with prior distribution π. Let δ(x) be an estimator where x=data. Then Bayes risk=$E_{\pi}(L(\delta, \theta))$. δ is Bayesian estimator that minimizes Bayes risk.
- Unsupervised: no labels/tags
- Hidden Markov Model (HMM)





HMM and POS

- Problem: Identify label sequence given word sequence
- Observed: word sequence (**w**). $|\mathbf{w}|=n$
- Hidden: POS sequence (**t**). #states=m
- Parameters:
  - Transition probabilities (θ_t) – Multinomial
  - Emission probabilities (ϕ_t) – Multinomial
  - Initial state distribution (π)
  - $\lambda = (\theta, \phi, \pi)$

Inference for HMMs

- Parameters:
  - Transition probabilities (θ_t) – Multinomial
  - Emission probabilities (ϕ_t) – Multinomial

$$\begin{aligned} \boldsymbol{\theta}_t &\mid \alpha &\sim \mathrm{Dir}(\boldsymbol{\alpha}) \\ \boldsymbol{\phi}_t &\mid \alpha' &\sim \mathrm{Dir}(\boldsymbol{\alpha}') \end{aligned}$$

- For experiments, they use uniform α and uniform α'.
- α controls sparsity of transition probabilities and α' controls sparsity of emission probabilities.
- α' →0
  - prior prefers models where each state emits as few words as possible
  - Situation: most words belong to a single POS

Bayesian estimation

- As against MLE/MAP, Bayesian estimation uses multiple values of parameters.
- Posterior does not have a closed form.

$$\underbrace{P(\theta|D)}_{\text{Posterior}} = \frac{\overbrace{P(\theta)}^{\text{Prior}}\overbrace{P(D|\theta)}^{\text{Likelihood}}}{\underbrace{\int P(\theta)P(D|\theta)d\theta}_{\text{Marginal Likelihood } (=P(D))}}$$

- Inference methods: EM, Variational Bayes (VB) estimation (approx), 4 types of Gibbs sampler (converge to true posterior)

Baum Welch (Forward-Backward/EM) Algorithm

- Compute forward and backward probabilities.

$$\alpha_k(t) = P(w_1,\ldots,w_k \mid t_k = t,\lambda) \quad \beta_k(t) = P(w_{k+1},w_{k+2},\ldots,w_n \mid t_k = t,\lambda)$$

- $\alpha_k(t)$ is the probability of observing a partial sequence of observables $w_1,\ldots w_k$ given state $t_k=t$ at time k, and λ
- $\beta_k(t)$ is the probability of observing a partial sequence of observables $w_{k+1},\ldots,w_n$ given state $t_k=t$ at time k, and λ
- Use dynamic programming to compute α and β

E Step

- Compute counts using forward and backward probabilities
- Let $n_{t't}$ be the probability of being in state *t* at time *k* and at state *t'* at time *k+1,* given λ and **w** sequence

$$n_{t',t}(k) = \frac{\alpha_k(t)\theta_{tt'}\phi_{t'}(w_{k+1})\beta_{k+1}(t')}{P(w \mid \lambda)} = \frac{\alpha_k(t)\theta_{tt'}\phi_{t'}(w_{k+1})\beta_{k+1}(t')}{\sum_{t=1}^{m}\sum_{t'=1}^{m}\alpha_k(t)\theta_{tt'}\phi_{t'}(w_{k+1})\beta_{k+1}(t')}$$

$$E[n_{t',t}] = \sum_{k=1}^{n-1} n_{t',t}(k)$$

- Let $n_t(k)$ be the probability of being in state *t* at time *k*, given *w*

$$n_t(k) = \sum_{t'=1}^{m} n_{t',t}(k) \qquad E[n_t] = \sum_{k=1}^{n-1} n_t(k) \qquad E[n'_{w,t}] = \sum_{k=1, w_k = w}^{n-1} n_t(k)$$

M step

- Use these counts to compute updated parameters.
- Iteratively re-estimates parameters.
- Converges to local maximum

$$\theta_{t'|t}^{(\ell+1)} = \mathrm{E}[n_{t',t}]/\mathrm{E}[n_t]$$
$$\phi_{w|t}^{(\ell+1)} = \mathrm{E}[n'_{w,t}]/\mathrm{E}[n_t]$$

- $n'_{w,t}$ is #times word w occurs with state t
- $n_{t',t}$ is #times state t' follows t
- n_t is #occurrences of state t
- $O(nm^2)$ time

Variational Bayes

- Aim: Find (**θ**,**ϕ**,**t**) that minimizes –log P(**w**)

$$-\log P(w)$$

$$= -\log \iiint Q(t,\theta,\phi).\frac{P(w,t,\theta,\phi)}{Q(t,\theta,\phi)} dt d\theta d\phi$$

$$<= -\iiint Q(t,\theta,\phi).\log \frac{P(w,t,\theta,\phi)}{Q(t,\theta,\phi)} dt d\theta d\phi$$

Jensen's inequality

$$-\log P(w)$$

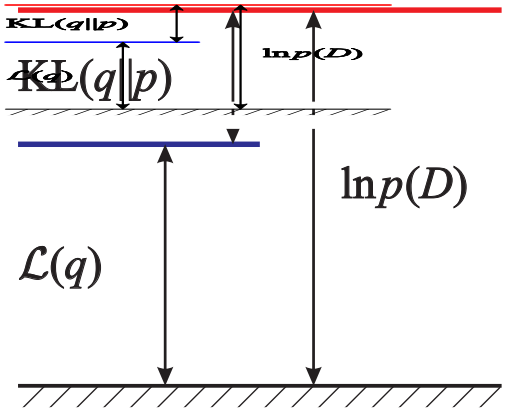
$$= -\iiint Q(t,\theta,\phi).\log \frac{P(w,t,\theta,\phi)}{Q(t,\theta,\phi)} dt d\theta d\phi - \iiint Q(t,\theta,\phi).\log \frac{Q(t,\theta,\phi)}{P(t,\theta,\phi \mid w)} dt d\theta d\phi$$

$$= -\iiint Q(t,\theta,\phi).\log \frac{P(w,t,\theta,\phi)}{Q(t,\theta,\phi)} dt d\theta d\phi - KL(q \mid p)$$

Variational free energy

Variational Bayes

- Find a $Q(t,\theta,\phi)$ that minimizes an upper bound to the negative log likelihood.
- Mean field assumption: local densities can be used to denote effects of global densities.
- Factorized model: $Q(t,\theta,\phi) = Q_1(t) \times Q_2(\theta,\phi)$
- Minimize the KL divergence between desired posterior distribution and factorized approximation.
- $O(nm^2)$



Variational Bayes

- If likelihood and prior belong to exponential family, VB is similar to Forward Backward Algorithm.
- E step is the same
- M step:

$$\tilde{\theta}^{(\ell+1)}_{t'|t} = f(\mathrm{E}[n_{t',t}] + \alpha) / f(\mathrm{E}[n_t] + m\alpha)$$

$$\tilde{\phi}^{(\ell+1)}_{w|t} = f(\mathrm{E}[n'_{w,t}] + \alpha') / f(\mathrm{E}[n_t] + m'\alpha')$$

$$f(v) = \exp(\Psi(v))$$

Smoothed counts

Digamma is first derivative of log gamma

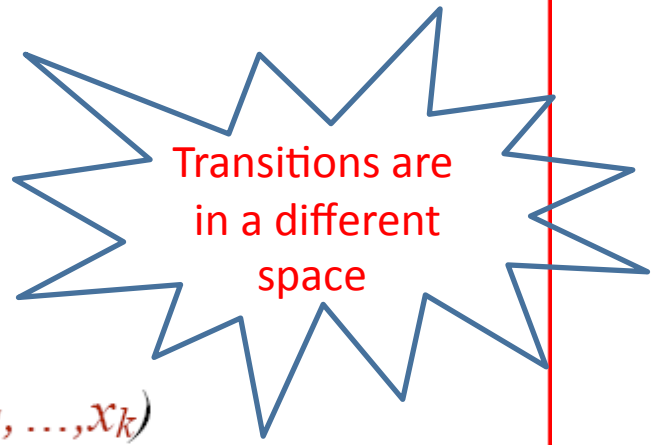
- m and m' are #word types and states.

Gibbs sampling

We will visit states according to transition probabilities $P(\boldsymbol{y}|\boldsymbol{x})$

That is, we will go from state $\boldsymbol{x} = (x_1, \ldots, x_k)$ to state $\boldsymbol{y} = (y_1, \ldots, y_k)$

For $i = 1 \ldots k$:
pick y_i by sampling from $P(Y_i \mid y_1, \ldots, y_{i-1}, x_{i+1}, \ldots, x_k)$



- We need all exact conditional distributions to estimate the joint probability distribution

MCMC sampling algorithms

- Produce a stream of samples from posterior distribution $P(\mathbf{t}|\mathbf{w}, \boldsymbol{\alpha})$
- 4 different Gibbs samplers:
  - Pointwise or blocked
  - Explicit or Collapsed
- Pointwise: Resamples a single state t_i (labeling a single word w_i) at each step. O(nm) per iteration.
- Blocked: Resamples labels for all of the words in a sentence at a single step. $O(nm^2)$ per iteration.
- Explicit: Samples $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ along with states $\mathbf{t}$
- Collapsed: $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are integrated out. Only $\mathbf{t}$ are sampled.

Pointwise explicit Gibbs sampler

- Resample **θ** and **ϕ** given state-to-state transition counts **n** and state-to-word emission counts **n'**
- Resample each state t_i given word w_i and neighboring states t_{i-1} and t_{i+1}

$$\begin{aligned} \boldsymbol{\theta}_t \mid \boldsymbol{n}_t, \boldsymbol{\alpha} &\sim \mathrm{Dir}(\boldsymbol{n}_t + \boldsymbol{\alpha}) \\ \boldsymbol{\phi}_t \mid \boldsymbol{n}'_t, \boldsymbol{\alpha}' &\sim \mathrm{Dir}(\boldsymbol{n}'_t + \boldsymbol{\alpha}') \\ \mathrm{P}(t_i \mid w_i, \boldsymbol{t}_{-i}, \boldsymbol{\theta}, \boldsymbol{\phi}) &\propto \theta_{t_i|t_{i-1}} \phi_{w_i|t_i} \theta_{t_{i+1}|t_i} \end{aligned}$$

Collapsed blocked Gibbs sampler

$$\theta^{\star}_{t'|t} = \frac{n_{t',t} + \alpha}{n_t + m\alpha}$$

$$\phi^{\star}_{w|t} = \frac{n'_{w,t} + \alpha'}{n_t + m'\alpha}$$

- Resample states for each sentence given **n** and **n'** for other sentences in the corpus.
- Following Metropolis-Hastings accept reject step, decide whether current state sequence be updated to t* or whether to keep current state sequence.
- High acceptance rates: 99%

Evaluation metrics

- Variation of information (VI): (lower the better)
  - VI=H(C)+H(C')-2I(C,C') where I(C,C')=H(C)-H(C|C')
  - The variation of information (VI) between two clusterings C (the gold standard) and C' (the found clustering) of a set of data points is a sum of the amount of information lost in moving from C to C', and the amount that must be gained.
  - Problem: Tagger that assigns all words the same POS has good VI
- Cross validation accuracy (higher the better)
  - Map each HMM state to the part-of-speech tag it co-occurs with most frequently (using train set), and use this mapping to map each HMM state sequence t to a sequence of part-of-speech tags (using validation set).
- Greedy 1-to-1 accuracy (higher the better)
  - At most 1 HMM state can be mapped to any POS tag.

Experiments

- 8 different combinations of hyper-parameters α and α' (0.0001 to 1)
- Data sets of different sizes (24K – 120K – 1174K words)
- Tag sets of different sizes (Noah Smith's 17 tag set, Penn Treebank tag set)
- Run each setting 10 times with at least 1000 iterations.

	All − 50	All − 17	120K − 50	120K − 17	24K − 50	24K − 17
EM	0.40527	0.43101	0.29303	0.35202	0.18618	0.28165
VB	0.46123	**0.51379**	0.34679	0.36010	0.23823	0.36599
$GS_{e,p}$	0.47826	0.43424	0.36984	0.44125	0.29953	0.36811
$GS_{e,b}$	0.49371	0.46568	0.38888	**0.44341**	0.34404	0.37032
$GS_{c,p}$	**0.49910***	0.45028	**0.42785**	0.43652	**0.39182**	**0.39164**
$GS_{c,b}$	0.49486*	0.46193	0.41162	0.42278	0.38497	0.36793

Figure 2: Average greedy 1-to-1 accuracy of state sequences produced by HMMs estimated by the various estimators. The column heading indicates the size of the corpus and the number of HMM states. In the Gibbs sampler (GS) results the subscript "e" indicates that the parameters θ and ϕ were explicitly sampled while the subscript "c" indicates that they were integrated out, and the subscript "p" indicates pointwise sampling, while "b" indicates sentence-blocked sampling. Entries tagged with a star indicate that the estimator had not converged after weeks of run-time, but was still slowly improving.

	All − 50	All − 17	120K − 50	120K − 17	24K − 50	24K − 17
EM	0.62115	0.64651	0.44135	0.56215	0.28576	0.46669
VB	0.60484	0.63652	0.48427	0.36458	0.35946	0.36926
$GS_{e,p}$	0.64190	0.63057	0.53571	0.46986	0.41620	0.37165
$GS_{e,b}$	**0.65953**	0.65606	0.57918	0.48975	0.47228	0.37311
$GS_{c,p}$	0.61391*	**0.67414**	**0.65285**	**0.65012**	**0.58153**	**0.62254**
$GS_{c,b}$	0.60551*	0.65516	0.62167	0.58271	0.55006	0.58728

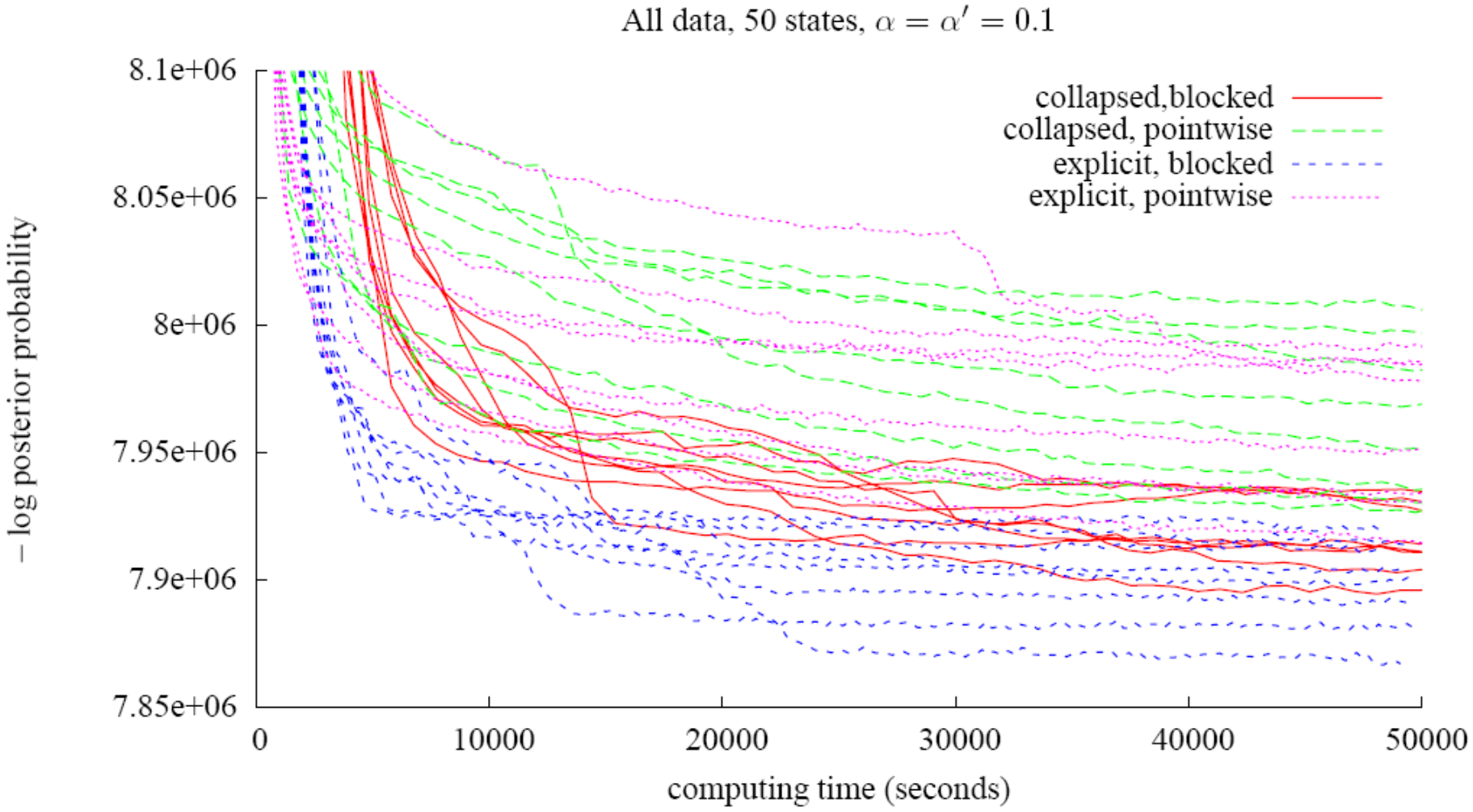
Figure 3: Average cross-validation accuracy of state sequences produced by HMMs estimated by the various estimators. The table headings follow those used in Figure 2.

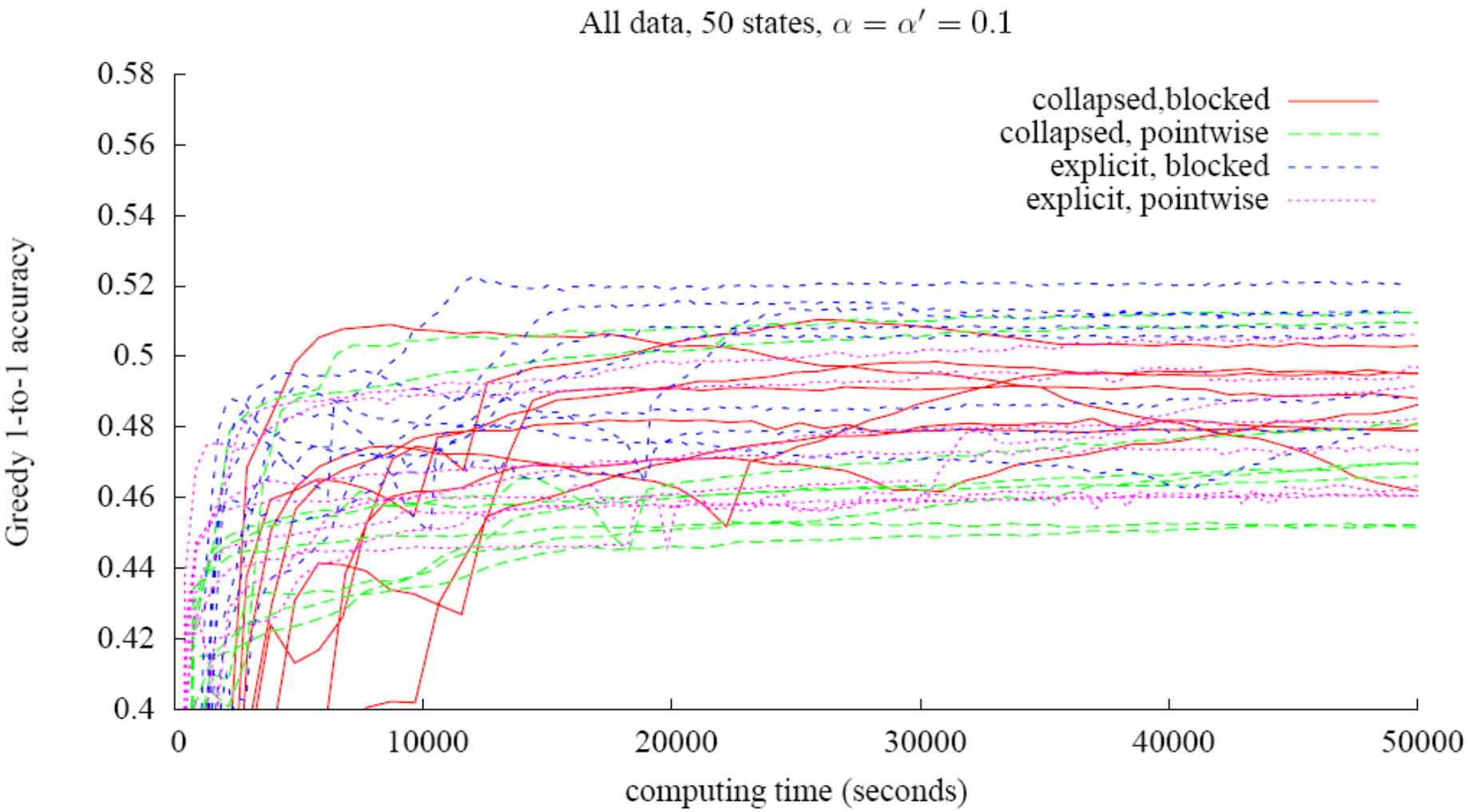
	All − 50	All − 17	120K − 50	120K − 17	24K − 50	24K − 17
EM	4.47555	3.86326	6.16499	4.55681	7.72465	5.42815
VB	4.27911	3.44029	5.00509	3.19670	4.80778	3.14557
$GS_{e,p}$	4.24919	3.53024	4.30457	3.23082	**4.24368**	3.17076
$GS_{e,b}$	4.04123	**3.46179**	4.22590	3.20276	4.29474	**3.10609**
$GS_{c,p}$	**4.03886**⋆	3.52185	**4.21259**	**3.17586**	4.30928	3.18273
$GS_{c,b}$	4.11272⋆	3.61516	4.36595	3.23630	4.32096	3.17780

Figure 4: Average Variation of Information between the state sequences produced by HMMs estimated by the various estimators and the gold tags (smaller is better). The table headings follow those used in Figure 2.

	All − 50	All − 17	120K − 50	120K − 17	24K − 50	24K − 17
EM	558	346	648	351	**142**	125
VB	**473**	**123**	**337**	**24**	183	**20**
$GS_{e,p}$	2863	382	3709	63	2500	177
$GS_{e,b}$	3846	286	5169	154	4856	139
$GS_{c,p}$	⋆	34325	44864	40088	45285	43208
$GS_{c,b}$	⋆	6948	7502	7782	7342	7985

Figure 5: Average number of iterations until the negative logarithm of the posterior probability (or likelihood) changes by less than 0.5% (smaller is better) per at least 2,000 iterations. No annealing was used.


All data, 50 states, $\alpha = \alpha' = 0.1$
collapsed,blocked
collapsed, pointwise
explicit, blocked
explicit, pointwise
– log posterior probability
8.1e+06
8.05e+06
8e+06
7.95e+06
7.9e+06
7.85e+06
0
10000
20000
30000
40000
50000
computing time (seconds)


All data, 50 states, $\alpha = \alpha' = 0.1$
collapsed,blocked
collapsed, pointwise
explicit, blocked
explicit, pointwise
Greedy 1-to-1 accuracy
0.58
0.56
0.54
0.52
0.5
0.48
0.46
0.44
0.42
0.4
0
10000
20000
30000
40000
50000
computing time (seconds)

Findings

- Point-wise samplers need O(m) steps per sample. EM, VB and sentence-blocked Gibbs sampler need $O(m^2)$ steps.
- On small datasets, all Bayesian estimators outperform EM (and to a lesser extent, VB).
  - Reasoning: Priors are imp when data is less. Also, approximation by VB would be inaccurate on small data.
- On large datasets, EM does well ito cross validation accuracy.
- VB converges faster. Larger α and α' cause faster convergence.
  - Reasoning: α and α' specify how likely the samplers are to consider novel tags and so influence sampler's mobility
- Blocked samplers converge faster than pointwise samplers. Explicit samplers are faster than collapsed ones.
- Pointwise samplers initially converge faster than blocked ones ➔ Hybrid strategy could be better.

Thanks!