Unsupervised HMM POS Tagger Comparison (Gao, Johnson)

John Wieting
CS 598

Unsupervised POS tagging

- Predict the tags for each word in a sentence
- 2 approaches used in this paper
  - Maximum likelihood $$L_{\boldsymbol{d}}(\boldsymbol{\theta}) = \mathrm{P}(\boldsymbol{d} \mid \boldsymbol{\theta})$$ $$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L_{\boldsymbol{d}}(\boldsymbol{\theta})$$
  - Bayesian $$\mathrm{P}(\boldsymbol{\theta} \mid \boldsymbol{d}) \propto \mathrm{P}(\boldsymbol{d} \mid \boldsymbol{\theta})\,\mathrm{P}(\boldsymbol{\theta})$$
    - Notice the prior which can bias the model
      - Use a Dirichlet prior to incorporate knowledge that words tend to only have few POS
    - Authors tend to not use MAP as they tend to prefer the full posterior as it incorporates the uncertainty of the parameters
    - No known closed form of posterior in most cases so MC and Variational Bayes approaches are used.

What is this paper about?

- Authors found that recent papers produced contradictory results about these Bayesian methods
- They study 6 algorithms
  - EM
  - Variational EM
  - 4 MCMC approaches
- Compare results on unsupervised POS tagging

HMM inference

- The parameters of an HMM are a pair of multinomials for each state t. The first specifies the distribution over states t' following state t and the second, the distribution over words w given t.
- Since this is a Bayesian model, priors are put on these multinomials. The authors use fixed and uniform Dirichlets for their simplification of inference.
  - These control the sparsity of the transition and emission probability distributions.
    - As they approach zero, the model strongly prefers sparsity (i.e. few words per tag)

$$t_i \mid t_{i-1} = t \sim \text{Multi}(\boldsymbol{\theta}_t) \qquad \boldsymbol{\theta}_t \mid \alpha \sim \text{Dir}(\boldsymbol{\alpha})$$
$$w_i \mid t_i = t \sim \text{Multi}(\boldsymbol{\phi}_t) \qquad \boldsymbol{\phi}_t \mid \alpha' \sim \text{Dir}(\boldsymbol{\alpha}')$$

Expectation Maximization

- Goal is to maximize the marginal log-likelihood

$$\ell(\theta) = \log p(x|\theta) = \log \sum_z p(x, z|\theta)$$
$$= \log \sum_z q(z|x, \theta) \frac{p(x, z|\theta)}{q(z|x.\theta)}$$
$$\geq \sum_z q(z|x, \theta) \log \frac{p(x, z|\theta)}{q(z|x, \theta)} \equiv F(q, \theta)$$

$$\textbf{E-step} : q^{(t+1)} = \arg\max_q F(q, \theta^{(t)})$$
$$\textbf{M-step} : \theta^{(t+1)} = \arg\max_\theta F(q^{(t+1)}, \theta)$$

$$q^{(t+1)} = p(z|x, \theta^{(t)})$$

$$\textbf{E-step} : \text{Compute } Q(\theta|\theta^{(t)}) = E_{p(z|x,\theta^{(t)})}[\log p(x, z|\theta)]$$
$$\textbf{M-step} : \theta^{(t+1)} = \arg\max_\theta E_{p(z|x,\theta^{(t)})}[\log p(x, z|\theta)]$$

ML EM in HMM

1. First compute forward and backward parameters which will be needed in M step

1. $\alpha_i(1) = \pi_i b_i(o_1)$
2. $\alpha_j(t+1) = \left[\sum_{i=1}^{N} \alpha_i(t) a_{ij}\right] b_j(o_{t+1})$
3. $p(O|\lambda) = \sum_{i=1}^{N} \alpha_i(T)$

1. $\beta_i(T) = 1$
2. $\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_j(t+1)$
3. $p(O|\lambda) = \sum_{i=1}^{N} \beta_i(1) \pi_i b_i(o_1)$

2. Then differentiate the Q function and maximize it subject to the constraint the probabilities sum to 1. Set to 0 and solve:

$$Q(\lambda, \lambda') = \sum_{q \in \mathcal{Q}} \log P(O, q|\lambda) P(O, q|\lambda')$$

$$Q(\lambda, \lambda') = \sum_{q \in \mathcal{Q}} \log \pi_{q_0} P(O, q|\lambda') + \sum_{q \in \mathcal{Q}} \left(\sum_{t=1}^{T} \log a_{q_{t-1} q_t} \right) p(O, q|\lambda') + \sum_{q \in \mathcal{Q}} \left(\sum_{t+1}^{T} \log b_{q_t}(o_t) \right) P(O, q|\lambda')$$

$$\frac{\partial}{\partial \pi_i} \left(\sum_{i=1}^{N} \log \pi_i p(O, q_0 = i|\lambda') + \gamma \left(\sum_{i=1}^{N} \pi_i - 1 \right) \right) = 0$$

3. Then you are done!

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

Variational EM

- In variational EM, we cannot represent our desired posterior in closed form. Thus we need to approximate it by minimizing the KL divergence between it and the posterior.
- This procedure works well for HMMs since the modifications to the E and M step turn out to be very minor. The updates in the M step are:

$$\begin{aligned}
\tilde{\theta}^{(\ell+1)}_{t'|t} &= f(\mathrm{E}[n_{t',t}] + \alpha) / f(\mathrm{E}[n_t] + m\alpha) \\
\tilde{\phi}^{(\ell+1)}_{w|t} &= f(\mathrm{E}[n'_{w,t}] + \alpha') / f(\mathrm{E}[n_t] + m'\alpha') \\
f(v) &= \exp(\Psi(v))
\end{aligned}$$

MCMC

- Samplers are either pointwise or blocked
  - pointwise = sample a single state ti corresponding to a particular word wi at each step (O(nm)).
  - blocked = resample all words in a sentence in a single step (O(nm^2)) using forward-backward algorithm varient.
- They are also either explicit or collapsed
  - explicit = sample HMM parameters (both theta and phi) as well as the states
  - collapsed = integrate out the HMM parameters and only sample the states
- In this paper all 4 possible variations are implemented and compared.

Pointwise and Explicit

- sample from the following distributions where nt is the state-to-state transition count and nt' is the state-to-word emission count.
- First sample the HMM parameters and then sample each state ti given the current word wi and the neighboring states ti and ti+1

$$\begin{aligned} \boldsymbol{\theta}_t \mid \boldsymbol{n}_t, \boldsymbol{\alpha} &\sim \mathrm{Dir}(\boldsymbol{n}_t + \boldsymbol{\alpha}) \\ \boldsymbol{\phi}_t \mid \boldsymbol{n}'_t, \boldsymbol{\alpha}' &\sim \mathrm{Dir}(\boldsymbol{n}'_t + \boldsymbol{\alpha}') \end{aligned} \quad (5)$$

$$\mathrm{P}(t_i \mid w_i, \boldsymbol{t}_{-i}, \boldsymbol{\theta}, \boldsymbol{\phi}) \propto \theta_{t_i|t_{i-1}} \phi_{w_i|t_i} \theta_{t_{i+1}|t_i} \quad (6)$$

Collapsed and Explicit

- Just sample from the following distribution:

$$\mathrm{P}(t_i|\boldsymbol{w}, \boldsymbol{t}_{-i}, \alpha, \alpha') \propto \left(\frac{n'_{w_i,t_i} + \alpha'}{n_{t_i} + m'\alpha'}\right) \left(\frac{n_{t_i,t_{i-1}} + \alpha}{n_{t_{i-1}} + m\alpha}\right) \left(\frac{n_{t_{i+1},t_i} + \mathrm{I}(t_{i-1} = t_i = t_{i+1}) + \alpha}{n_{t_i} + \mathrm{I}(t_{i-1} = t_i) + m\alpha}\right)$$

Pointwise and Blocked

- Here we are resampling an entire sentence
- How?
  - First resample HMM parameters (using equations from pointwise and explicit sampler), then use forward-backward algorithm to sample a structure.

$$\xi_{ij}(t) = \frac{p(Q_t = i, Q_{t+1} = j, O|\lambda)}{\varepsilon_{ij}(t) \propto p(Q_{t+1} = j|Q_t = i, O, \lambda)} = \frac{\alpha_i(t) a_{ij} b_j(o_{t+1}) \beta_j(t+1)}{) a_{ij} b_j(o_{t+1}) \beta_j(t+1)}$$

  - Once done, we can update the counts to be used for the sampling step in the next iteration.

Collapsed and Blocked

- In this model, we again iterate through the sentences resampling the states for each sentence conditioned on n (state-to-state) and n' (state-to-word).
  - Need to first compute parameters of a proposal HMM

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

- Then a structure is sampled using the dynamic algorithm mentioned on the slide.
- The motivation for the proposal distribution is that we want to sample from

$$\mathrm{P}(t_i|w_i, \mathbf{t}_{-i}, \alpha) = \frac{\mathrm{P}(w_i|t_i)\mathrm{P}(t_i|\mathbf{t}_{-i}, \alpha)}{\mathrm{P}(w_i|\mathbf{t}_{-i}, \alpha)}$$

Collapsed and Blocked

- However that denominator is tough to compute. So a Hasting's Sampler is used to sample from the desired distribution. The sample distribution chosen was to use the distribution whose parameters are $\mathbb{E}[\theta|\mathbf{t}_{-i}, \alpha]$

$$A(s, s') = \min\left\{1, \frac{\pi(s')Q(s|s')}{\pi(s)Q(s'|s)}\right\}$$

$$\begin{aligned} A(t_i, t_i') &= \min\left\{1, \frac{\mathrm{P}(t_i'|w_i, \mathbf{t}_{-i}, \alpha)\mathrm{P}(t_i|w_i, \theta')}{\mathrm{P}(t_i|w_i, \mathbf{t}_{-i}, \alpha)\mathrm{P}(t_i'|w_i, \theta')}\right\} \\ &= \min\left\{1, \frac{\mathrm{P}(t_i'|\mathbf{t}_{-i}, \alpha)\mathrm{P}(t_i|w_i, \theta')}{\mathrm{P}(t_i|\mathbf{t}_{-i}, \alpha)\mathrm{P}(t_i'|w_i, \theta')}\right\} \end{aligned}$$

Evaluation

- How to evaluate?
  - We need to somehow map a system's states to the gold standard states
  - Variation of Information
    - information theoretic measure that measures the difference in information between two clusters
    - unfortunately this approach allows a tagger that assigns each word the same tag to perform well.
  - Mapping approaches
    - map each hmm state to the most common POS tag occurring in it.
      - Issue with this approach is that it rewards HMMs with large amounts of states

Evaluation

- More mapping approaches
  - Split gold data set and do the state mapping on one half and use the other half for evaluation (cross validation approach)
  - Insist that at most one HMM state can be mapped to a particular POS tag
    - Used greedy algorithm to match states to tags until it runs out of states/tags. Unassigned states/tags are left unassigned.

Results

- In their experiments, the authors vary the number of tags and the size of the corpus.
- For each model they optimize the two hyperparameters over a range of values ranging from 0.0001 to 1 and report the results for the best set for that model.
- As expected, on small data sets, the prior seems to play a more important role and so the MCMC approaches do better than EM and VB (which has a worse approximation with smaller amounts of data).
- On larger data sets the results evened out though.
- In terms of convergence time, blocked samplers were faster than pointwise and explicit were faster than collapsed.

Results

	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.

Results

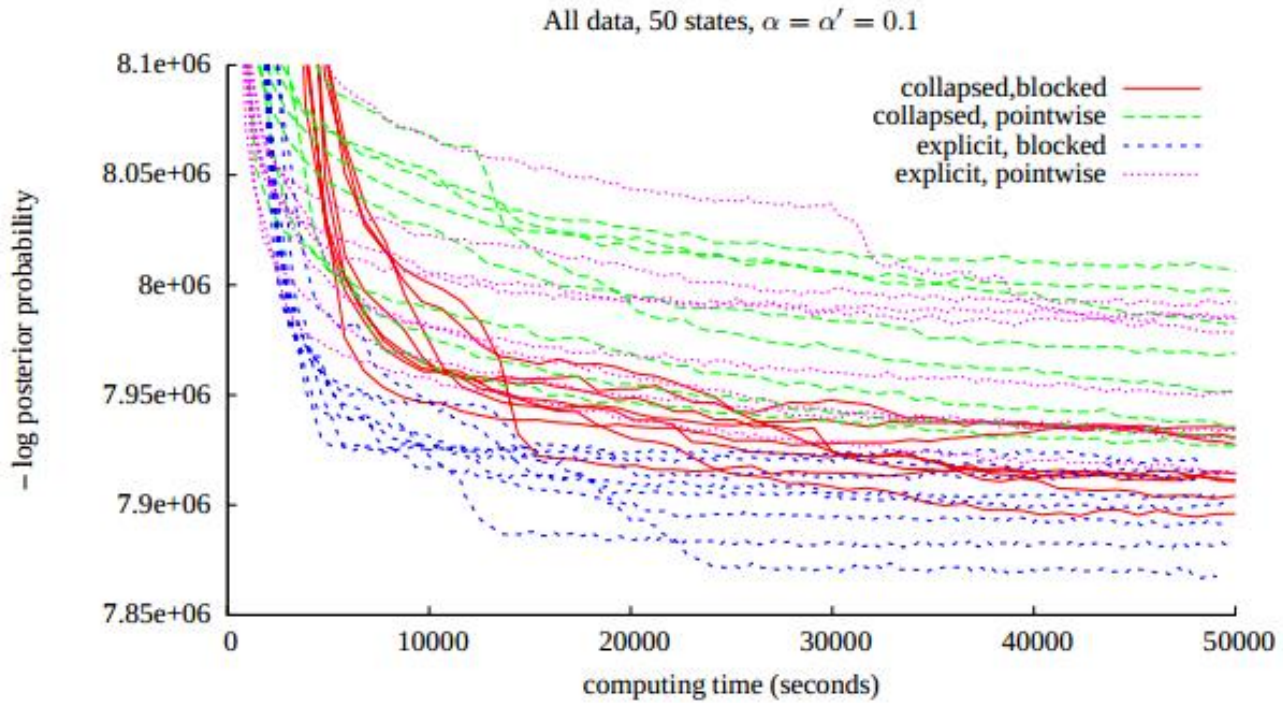
	All − 50	All − 17	120K − 50	120K − 17	24K − 50	24K − 17
EM	4.47555	3.86326	6.16499	4.55681	7.72465	5.42815
VB	4.27911	3.44029	5.00509	3.19670	4.80778	3.14557
$GS_{e,p}$	4.24919	3.53024	4.30457	3.23082	**4.24368**	3.17076
$GS_{e,b}$	4.04123	**3.46179**	4.22590	3.20276	4.29474	**3.10609**
$GS_{c,p}$	**4.03886**⋆	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.

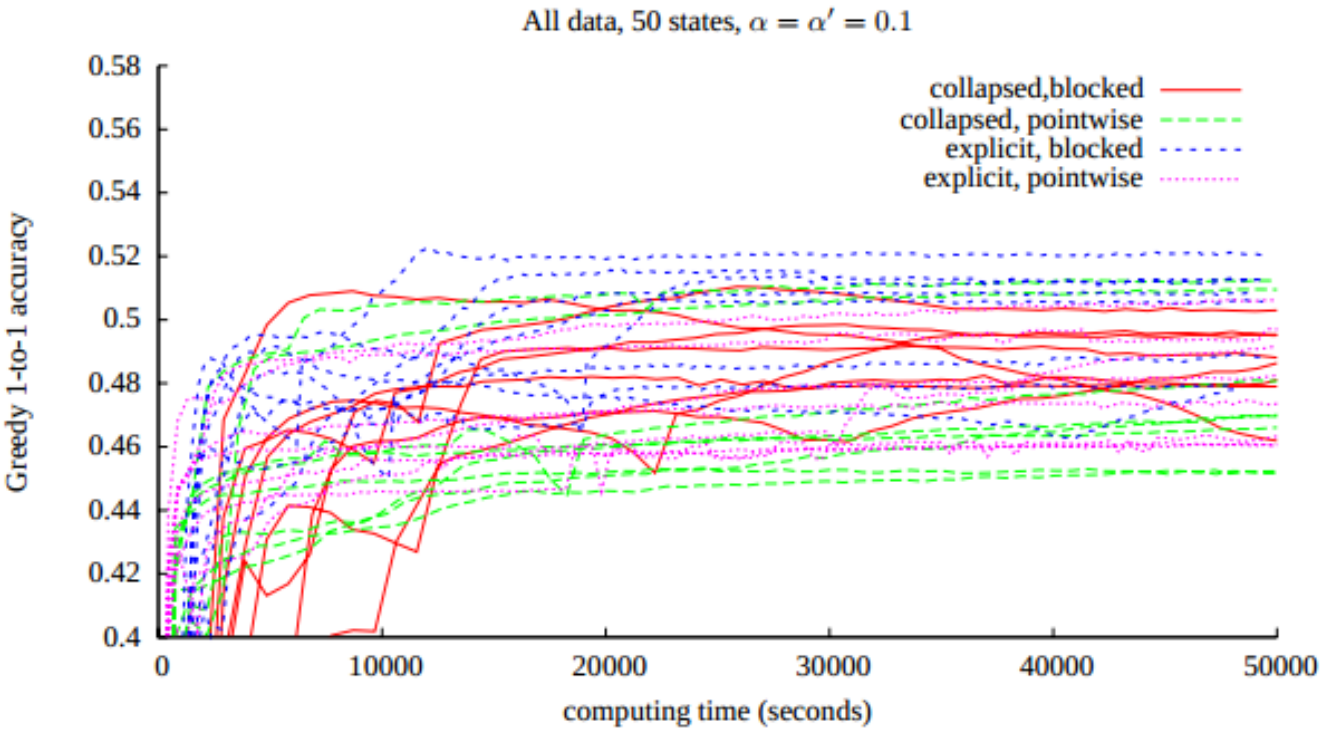
	All − 50	All − 17	120K − 50	120K − 17	24K − 50	24K − 17
EM	558	346	648	351	**142**	125
VB	**473**	**123**	**337**	**24**	183	**20**
$GS_{e,p}$	2863	382	3709	63	2500	177
$GS_{e,b}$	3846	286	5169	154	4856	139
$GS_{c,p}$	⋆	34325	44864	40088	45285	43208
$GS_{c,b}$	⋆	6948	7502	7782	7342	7985

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.

Results


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)

Results


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)

Summary

- This paper compared the performance of 5 different Bayesian approaches and 1 ML approach to unsupervised POS tagging using HMMs.
- The comparison spanned different numbers of hidden states and different amounts of training data
- Gibbs sampling approaches seemed to perform the best however their advantage decreased as the data sets increased in size
- VB was the fastest Bayesian model