DD2447 Project: Gibbs Sampler

Alden Coots

22 January 2013

The following generative model generates K sequences of length N : $\mathbf{s}_1, \cdots, \mathbf{s}_K$ where $\mathbf{s}_i = s_{i,1}, \cdots, s_{i,N}$. All sequences are over the alphabet $[M]$. Each of these sequences has a "magic" word of length w hidden in it. The rest of the sequence is background.

First for each i, a start position r_i for the magic word is sampled uniformly from $[N-w+1]$. Then the j:th positions in the magic words are sampled from $q_j(x)$, which is $\mathrm{Cat}(\mathbf{x}|\theta_j)$ where θ_j has a $\mathrm{Dir}(\theta_j|\alpha)$ prior. All other positions in the sequences are sampled from the background distribution q(x), which is $\mathrm{Cat}(\mathbf{x}|\theta)$ where θ has a $\mathrm{Dir}(\theta|\alpha')$ prior. This can be done with a call to `generate(num_seq, seq_length, alphabet, m_word_length, m_word_param, background_param)`

C	C	C	G	G	C	G	A	A	A
G	C	A	C	G	G	G	T	G	G
G	C	C	T	A	T	A	C	A	G
A	G	T	A	G	A	C	T	G	A
G	G	C	C	T	C	T	A	A	A
G	A	A	C	C	C	C	G	G	G
A	G	G	C	G	G	G	A	C	C
A	A	G	A	G	A	C	C	G	G
A	C	C	G	T	A	C	C	C	A
G	A	G	C	G	G	G	A	G	T
C	T	C	C	C	C	T	G	G	A
A	T	G	A	G	A	G	G	A	A
G	C	G	C	G	G	G	A	T	A
G	C	C	T	G	G	A	G	C	A
A	G	A	C	G	C	T	G	C	A

Table 1: 15 sequences of length 10 with hidden word of size 5 generated from alphabet [A, C, G, T] with $\alpha = [1, 2, 2, 1]$ and $\alpha' = [1, 1, 1, 1]$

The posterior over starting positions is then estimated using a collapsed Gibbs sampler, by repeatedly updating all magic word starting positions, R, by sampling $P(r_z|\mathcal{D}_{-z}, R_{-z}, \alpha)$ for each sequence s_z from s_1 to s_K in random order. This is accomplished by passing the result of `generate(...)` as the `pos_sequences` parameter to `gibbssample(num_iters, pos_sequences, alphabet, m_word_length, m_word_param, background_param)`.

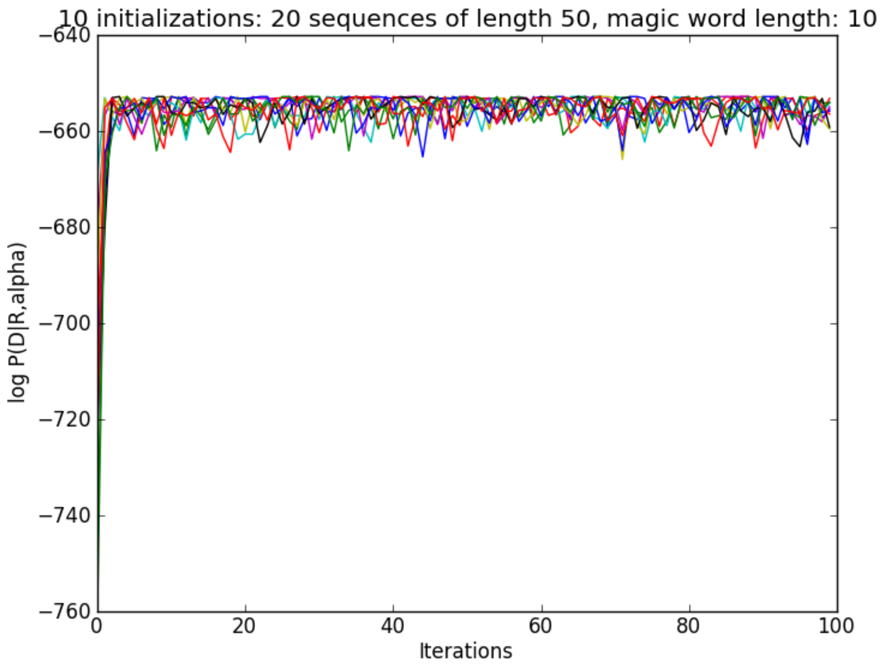
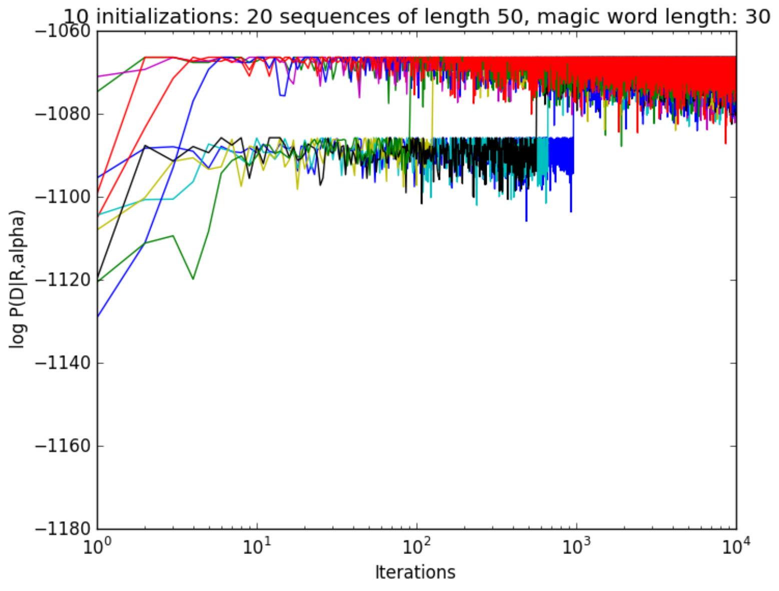
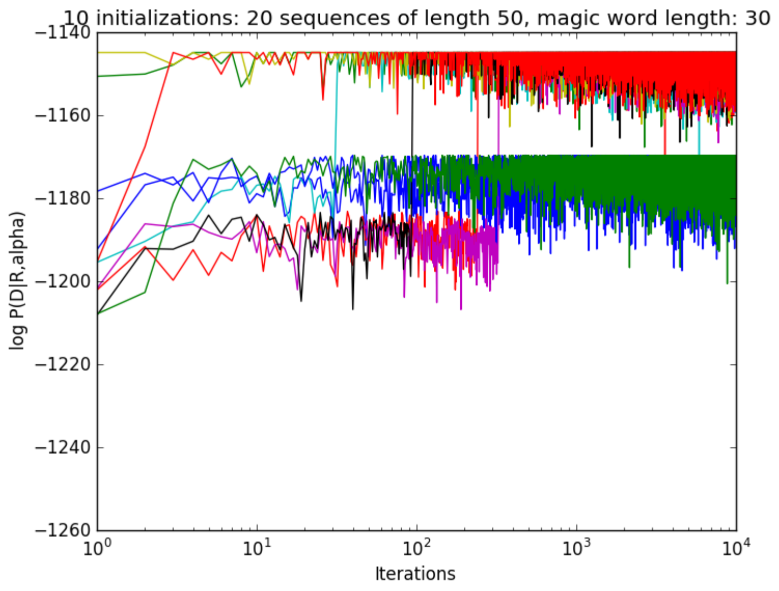


Figure 1: Graph showing convergence behavior of 10 random initializations of starting positions for a 4 letter alphabet with $\alpha = [1, 2, 2, 1]$ and $\alpha' = [1, 1, 1, 1]$

As the number of iterations grows large, the chance that the algorithm has converged becomes greater; however, at any given number of iterations, there is no guarantee that the stationary distribution has been reached. As evidenced by many of the figures, the distribution often gets stuck in local modes. By calculating the posterior over starting positions with several different random initializations, the likelihood of determining the correct starting positions of the magic words is increased.

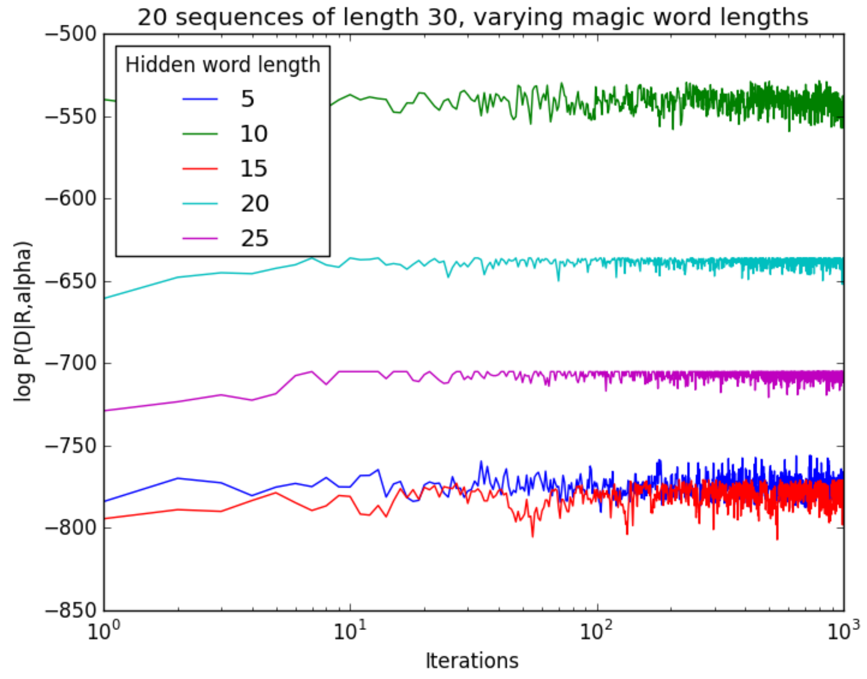


(a) All initializations are likely converged



(b) Blue and green initializtions get stuck in local maxima. The black, red, and purple manage to escape their local modes

Figure 2: 4 letter alphabet with $\alpha = [1, 2, 2, 1]$ and $\alpha' = [1, 1, 1, 1]$



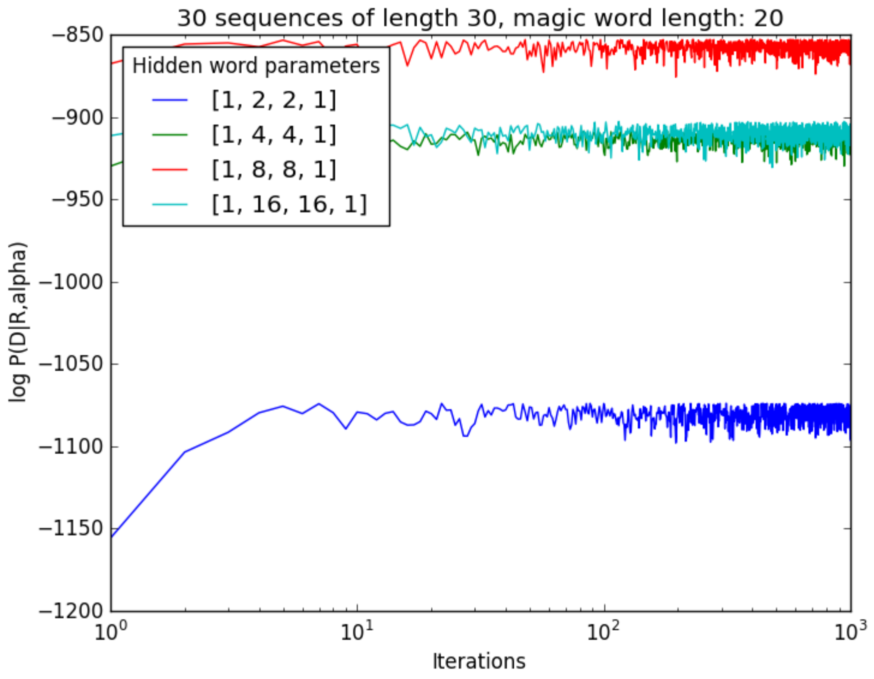
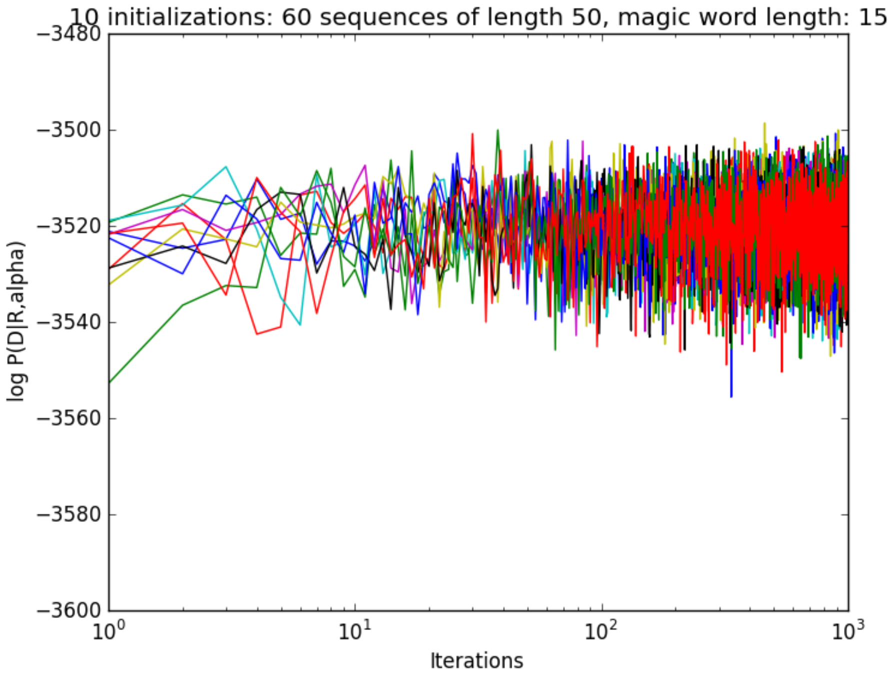
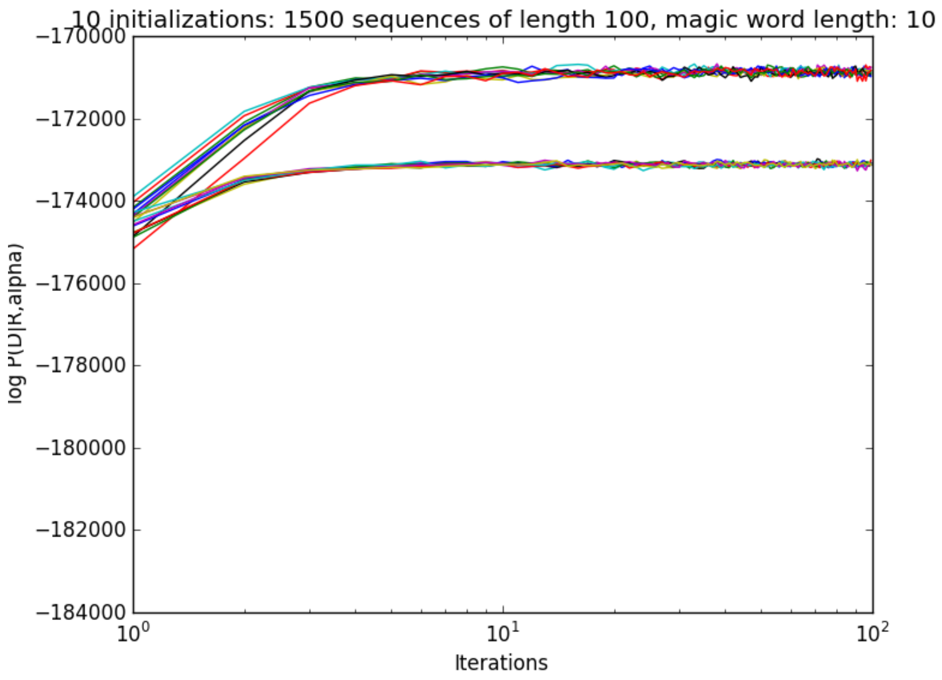


Figure 3: Moderate changes in hidden word length and hidden word parameter value appear to have no significant effect on the number of iterations to possible convergence.



(a) 4 letter alphabet with $\alpha = [34, 2, 6, 21]$ and $\alpha' = [2, 12, 9, 3]$. Altering both α and α' makes convergence more difficult.



(b) 4 letter alphabet with $\alpha = [34, 2, 6, 21]$ and $\alpha' = [1, 1, 1, 1]$. Increasing the number of sequences helps to speed up convergence, although some initializations get easily stuck in local modes