HDP:Dirichlet-Multinomial

Donglai Wei

2010.5.12

0)Problem

Though ME algorithm can find an "almost good" log probability, the clustering result is far from truth. The hope may lie in the "communicating power" among the dishes during **Initialization** and **Searching Tricks**.

Hyperparameters:$\alpha = 0.5, \gamma = 1.5, \phi_0 = 0.2$

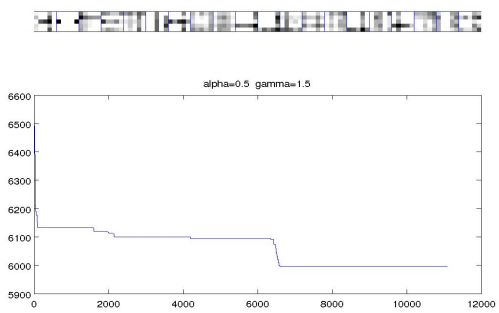

Figure 1: ME result

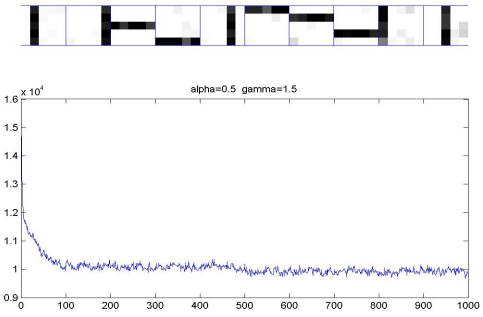

Figure 2: Gibbs

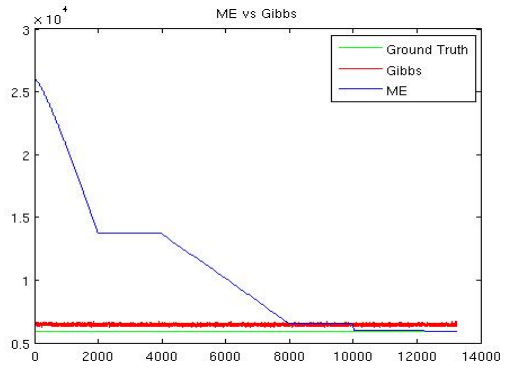

Figure 3: ME vs Gibbs

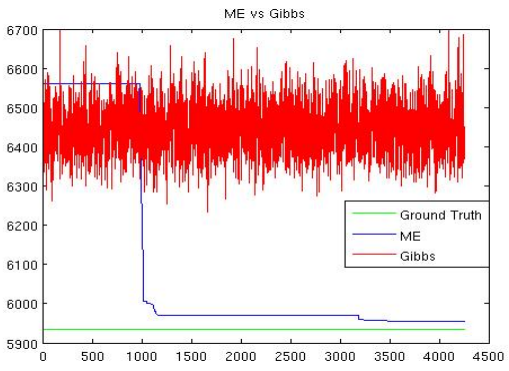

Figure 4: ME vs Gibbs

1) Formula

W:number of unique words
$n^w_{..k}$number of occurence of word w in dish k

$-logp(x|z,\lambda)$
=
(Likelihood-term)$\sum_{k=1}^{K}[log(\frac{\Gamma(n_{..k}+W\phi_0)}{\Gamma(W\phi_0)}) + \sum_{w=1}^{W} log(\frac{\Gamma(\phi_0)}{\Gamma(\phi_0+n^w_{..k})})]$
+
(Allocation-term)$\underline{\sum_{j=1}^{J}\{log\frac{\Gamma(n_{j..}+\alpha)}{\Gamma(\alpha)} - \sum_{t=1}^{m_{j.}}[log(\Gamma(n_{jt.}) + log\alpha]\} + log\frac{\Gamma(T+\gamma)}{\Gamma(\gamma)} - \sum_{k=1}^{K}[log(\Gamma(m_{.k}) + log\gamma]}$

=
(t-term)$\underline{log\frac{\Gamma(T+\gamma)}{\Gamma(\gamma)} + \sum_{j=1}^{J}\{log\frac{\Gamma(n_{j..}+\alpha)}{\Gamma(\alpha)} - \sum_{t=1}^{m_{j.}}[log(\Gamma(n_{jt.}) + log\alpha]\}}$

+(k-term)$\sum_{k=1}^{K}[log(\frac{\Gamma(n_{..k}+W\phi_0)}{\Gamma(W\phi_0)}) + log(\Pi_{w=1}^{W}\frac{\Gamma(\phi_0)}{\Gamma(\phi_0+n^w_{..k})}) - \underline{log(\Gamma(m_{.k}) - log\gamma]}$

2) Initialization?

i)Extreme Case

Bottom-up Initialization: All customers are assigned to different tables with different dishes.
Top-down Initialization: Every Restaurant has only one table with the same dish.

We know they are of no good since they don't capture potential mixture components.

Indicated by the toy bar example, with these initialization, ME algorithm quickly get stuck at the configuration: 1 table for each restaurant, where the t-term$\sim$ 200,k-term$\sim$ 6,000 while for the ground truth:t-term$\sim$ 3,000,k-term$\sim$ 3,000

ii)Somewhere in Between

Hoping that some restaurant may have only one mixture component,we have the following initializaiton:

For each of the J restaurants independently:

1. Make J-1 tables serving J-1 dishes, where each dish is informed by all the data from one of the other restaurants
2. Search over assignments t_{ji} of customers to these tables (keep k_{jt} fixed and don't allow the option of making a new table&dish not shared with any other restaurant)
3. Store the at most J-1 tables with at least one customer (hopefully this will be a number bigger than one but much less than J-1)

But the problems are:

1. **It may be equivalent to simply assigning customers with the same type(word) to the same table.**
   Suppose we have 40 Restaurants, each has 50 customers.
   For the 50 customers in Restaurant 1, they have 39 tables(each formed by a restaurant) to choose from.
   For the first customer(say with word w_0), t-term is the same for assigning it any of the 39 tables(since n_{jt}=50 for all in this case)
   The difference among the k-term happens at$log\frac{1}{\Gamma(\phi_0+n_{..k}^{w_0})}$ which is concave(bigger $n_{..k}^{w_0}$,bigger decrease)
   Thus assign the first customer to the table with the highest count of word w_0 will mostly decrease the Free Energy(Negative log Probability).

2. In the toy bar data, unfortunately, the highest count of a certain word happens at the cross of two bars in a restaurant, which still does not capture any correlation among the word. As a result, in every restaurant, each different type of word is assigned to a different table, which doesnot incorporate any potential topic information.

iii)Experiment

1)I tried the new Initialization, which forces dishes to have only single word.
2)I tried to initialize with the result from Gibbs Sampling, which works well.
(The "Ground Truth" is made up...since there are exponentially many configurations even though we know the true mixture components)

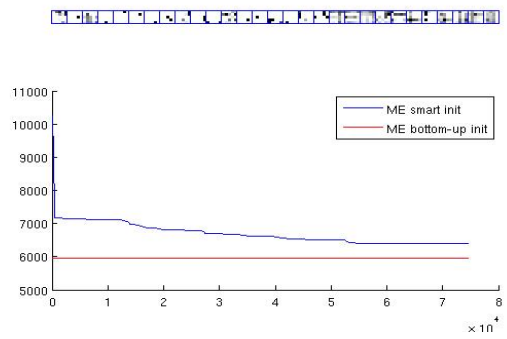

Figure 5: somewhere in between initializaiton

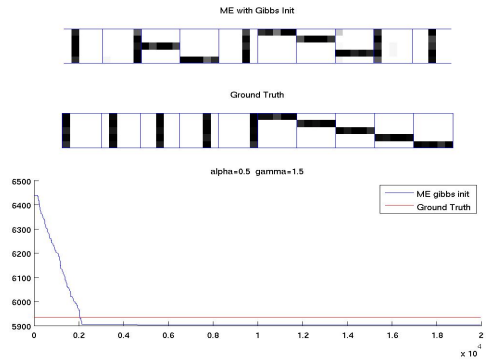

Figure 6: Gibbs Initialization

3) Searching Scheme?

What do we really want?(log probability is definitely not the answer)

Take the toy bar data as an example.
Given 25 words,40 Restaurants each of which is a mixture of bars.

Clue1: Each dish has its own vocabulary,one of $2^{25}-1$ nonempty subsets of (1,...,25). We want these dishes to be the true mixture component(bars) to construct each restaurant

Clue2: In the object function, most terms can be organized into "partition cost".e.g. $\frac{\Gamma(n_{..k}+W\phi_0)}{\Pi_1^W\Gamma(\phi_0+n_{..k}^w)}$

is the cost to divide dish vocabulary into each word. The bigger the partition component, the smaller the cost

Want to minimize:$-logp(x|z,\lambda)$
=
(Smaller Dish vocabulary term)$log[\Pi_{k=1}^{K}(\frac{\Gamma(n_{..k}+W\phi_0)}{\Pi_1^W\Gamma(\phi_0+n_{..k}^w)})] - Tlog\alpha - Klog\gamma$
(Bigger Dish vocabulary term)$+Klog\frac{\Gamma(\phi_0)^W}{\Gamma(W\phi_0)} + log\frac{\Gamma(T+\gamma)}{\Pi_{k=1}^K\Gamma(m_{.k})} + log\Pi_{j=1}^J\frac{\Gamma(n_{j..})}{\Pi_{t=1}^{m_j}\Gamma(n_{jt.})}$
(constant-term)$-log\Gamma(\gamma) - Jlog(\Gamma(\alpha)) + log\Pi_{j=1}^J\frac{\Gamma(n_{j..}+\alpha)}{\Gamma(n_{j..})}$

Fixing α,γ:

1. Smaller Dish vocabulary term:
   (a) $\frac{\Gamma(n_{..k}+W\phi_0)}{\Pi_1^W\Gamma(\phi_0+n_{..k}^w)}$ wants the dish to have bigger partition component (concentrated on some of the words).
   e.g. $\frac{\Gamma(10)}{\Gamma(5)\Gamma(5)} << \frac{\Gamma(10)}{\Gamma(1)...\Gamma(1)}$
   (b) Consequently, smaller dish vocabulary may need more dishes K and more tables T to explain each Restaurant:$-Tlog\alpha - Klog\gamma$
2. Bigger dish vocabulary term:
   (a) $log\Pi_{j=1}^J\frac{\Gamma(n_{j..})}{\Pi_{t=1}^{m_j}\Gamma(n_{jt.})}$ wants each restaurant has only one table, thus a big dish vocabulary.
   (b) $log\frac{\Gamma(T+\gamma)}{\Pi_{k=1}^K\Gamma(m_{.k})}$ wants smaller number of dishes:
   e.g. $\frac{\Gamma(10)}{\Gamma(5)\Gamma(5)} << \frac{\Gamma(10)}{\Gamma(1)...\Gamma(1)}$
   (c) Consequently, bigger dish vocabulary may need less dishes K to explain each Restaurant:$Klog\frac{\Gamma(\phi_0)^W}{\Gamma(W\phi_0)}$