ME p.k. Gibbs Sampling

Donglai Wei

2010.6.14

0)Outline

1. ME: From Split to Decompose
2. Gibbs: when to fail

1) ME for Dirichlet-Multinomial

i) Formula

W:number of unique words
$n_{..k}$number of customers in dish k
$n^{w}_{..k}$number of occurence of word w in dish k
$n_{j..}$number of customers in in Restaurant j
$n_{jt.}$number of customers in table t in Restaurant j
$m_{..}$number of tables in total
$m_{.k}$number of tables in dish k

$$-P = -logp(x, z|\lambda)$$

=

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

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

=

(z-term)+ $\sum_{k=1}^{K}[log(\frac{\Gamma(n_{..k}+W\phi_0)\Gamma(\phi_0)^{W}}{\Gamma(W\phi_0)\Pi_1^{W}\Gamma(\phi_0+n^{w}_{..k})\Pi_{t^*=1}^{m_{.k}}\Gamma(n_{.t^*k})})]$

(m-term)$-m_{..}log\alpha + log\frac{\Gamma(m_{..}+\gamma)}{\Pi_{k=1}^{K}\Gamma(m_{.k})}$

(constant-term)$-log\Gamma(\gamma) + \sum_{j=1}^{J} log\frac{\Gamma(n_{j..}+\alpha)}{\Gamma(\alpha)}$

=

(table-res-term)$\sum_{j=1}^{J} log\frac{1}{\Pi_{t=1}^{m_{j.}}\alpha\Gamma(n_{jt.})}$

(table-dis-term)$\sum_{k=1}^{K} log\frac{\Gamma(n_{..k}+W\phi_0)\Gamma(\phi_0)^{W}}{\Pi_1^{W}\Gamma(\phi_0+n^{w}_{..k})\gamma\Gamma(W\phi_0)}$

(table-num-term)$log\frac{\Gamma(m_{..}+\gamma)}{\Pi_{t=1}^{m_{j.}}\Gamma(m_{.k})}$

(constant-term)$-log\Gamma(\gamma)+\sum_{j=1}^{J} log\frac{\Gamma(n_{j..}+\alpha)}{\Gamma(\alpha)}$

ii) Comparison:Split v.s. Decompose

a)Intuition

i)Intuitively, Split-Merge is designed for Gaussian case, where a cluster should be defined as group of points close to each other.
ii)It is natural to do split move in the Euclidean space, where it is kind of "transive".(e.g. if cluster 1 should be splitted into three clusters, it may still be better off if we first split it into 2 clusters.)

But for Dirichlet-Multinomial case:
i) the Sample Space itself is discrete.
ii)The aim is to find a reasonable size of topics to **explain** the raw documents.

b)Strategy

The Goal of ME algorithm is to search for the best assignment variable $\vec{z}$ that maximize the log probability P.

The basic idea is to search **"locally"**,changing the config of a table/restaurant/dish conditioned on others fixed.

1. Given other Restaurants fixed, want to find the best Restaurant j's Config:
   1)Split:2-means++(sampling dishes) $m_{j.} \rightsquigarrow 2 * m_{j.}$ +TKM
   2)Decompose: samples the dish component+TKM
2. Given other dishes fixed, want to find the best Dish k's Config:
   1)Split:2-means++(sampling dishes) $m_{.k} \rightsquigarrow m_{.k}+1$ +TKM
   2)Decompose: samples the dish component+TKM

Obviously, the Decompose move is much more flexible.

c)Problem with Split

1) Senario 1:(Noisy Dishes) If several dishes(mixture of 2 bars) share some bars, it will be hard to figure out the true bar by split dishes **sequentially**.
2) Senario 2:(Multiple Bars) If one dish is made of multiple bars, the **two** new splitted dishes won't be explained well by other bars.
Solution:(which is kind of nasty even for toy data)
1)Split-Dish-All
2)Split-Dish-Further

iii) Decompose move

a)Decompose Restaurant

(i) Decompose Restaurant j:

(a) Iterate until no customers are left

(b) For each dish k:calculate the weight of forming a table(with dish k) with the overlapped words in the restaurant

(c) Sample the table according the weight and make it

(ii) TKM

NOTE::
During Initialization: TKM enables new table/dish;weight is ΔP
During Search:TKM doesn't allow new table/dish;weight is $\Delta\frac{\Gamma(m_{..}+\gamma)}{\Pi_1^W\Gamma(\phi_0+n^w_{..k})}$

b)Decompose Dish

(i) Decompose each table(into 1 or 2 tables) in dish k:

(a) Remove the table t from the restaurant

(b) For each dish K$\neq$k:calculate the weight of forming a table(with dish k) with the overlapped words in the table

(c) Sample the table according the weight and make it

(d) make the rest of the words in the table back to table t.

(e) Local Search table

(ii) Split-k

(iii) TKM

NOTE::
Decompose Table:weight is $\Delta\frac{\Gamma(m_{..}+\gamma)}{\Pi_1^W\Gamma(\phi_0+n^w_{..k})}$
During Initialization: TKM enables new table/dish
During Search:TKM doesn't allow new table/dish

iv) Experiment on 200 Restaurants(5*5)

Algorithm:
while until P doesn't increase:
1)Decompose Restaurant(Initialization)
2)Decompose Dish(Initialization)
while until P doesn't increase:
3)Decompose Restaurant(Search)
4)Decompose Dish(Search)
End
End

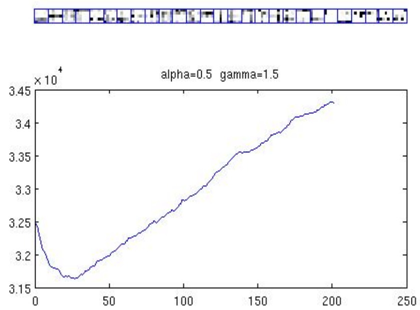

Figure 1: 1)DR Initialization

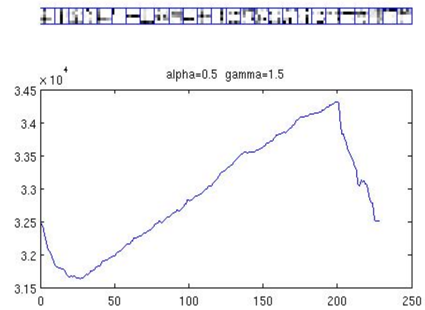

Figure 2: 2)DD Initialization

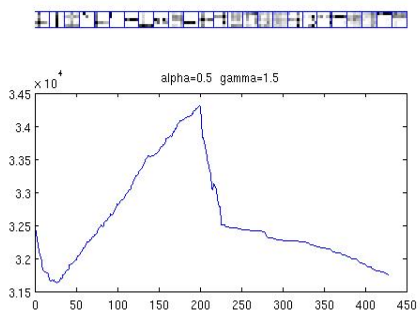

Figure 3: 3)DR Search

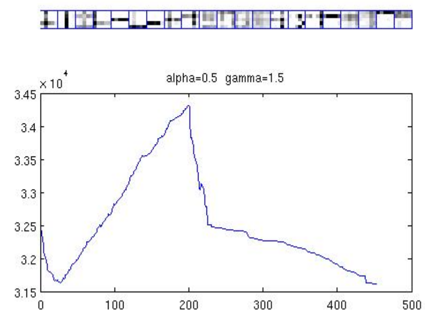

Figure 4: 4)DD Search

2) Gibbs

Problems for Gibbs:

1. Hard to decide when to stop
2. Quick to find the bars, but hard to purify them with large moves

i)Comparison on 200 Restaurants(5*5)

From Teh's package 1.0, test Gibbs sampling methods: crf,beta,block.

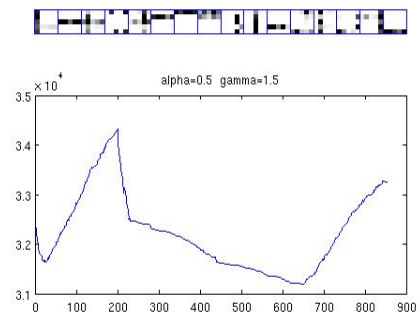

Figure 5: 1)DR Initialization

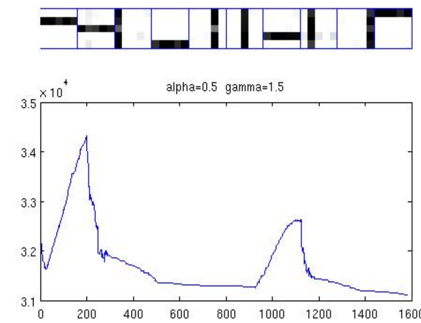

Figure 6: Finally...

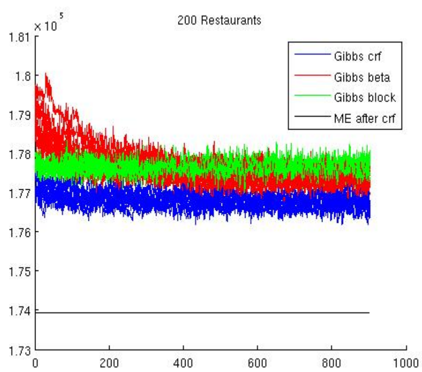

Figure 7: 1)DR Initialization

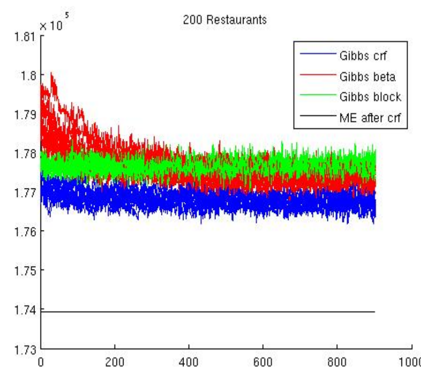

Figure 8: Finally...