Pseudo-code for Split and Merge

Donglai Wei

2010.4.21

Goal

find higher log-probability P=log(p(x$|\lambda, \vec{z}$))

Code

While no more changes of table assignment and dish assignment can increase P:

(I)

$$a = rand([0,1]) \Rightarrow \begin{cases} Go\ to\ Step\ II:\ Split\ tables & a \in [0, \frac{1}{4}] \\ Go\ to\ Step\ III:\ Merge\ tables & a \in [\frac{1}{4}, \frac{2}{4}] \\ Go\ to\ Step\ IV:\ Split\ dishes & a \in [\frac{2}{4}, \frac{3}{4}] \\ Go\ to\ Step\ V:\ Merge\ dishes & a \in [\frac{3}{4}, 1] \end{cases}$$

(II) Split tables:

(1) Randomly pick a Restaurant R, which has m tables.

(2) Randomly pick a table t in R, which has n customers.

(3) if(n≠1) Do 2-means++ (current: K dishes,table assignment:z_{t0},dish assignment:z_{k0})

(a) Initialization:

Randomly pick a customer C1 in from table t to form a new table (m+1) with a new dish(K+1) $\Rightarrow (z_{t1}, z_{k1})$

For ww=randperm(customers from table t except C1)
z'_{t1}: change z_{t1} by assigning customer ww to the new table (m+1)
weight(ww)=max$\{log(p(x|\lambda, z_{t1})) - log(p(x|\lambda, z'_{t1})), 0\}$ (more decrease,more weight)
End

Randomly sample another customer C2 from table t according to weight,form a new table (m+2) with a new dish(K+2)$\Rightarrow (z_{t2}, z_{k2})$

For ww=randperm(customers from table t except C1,C2)
z'_{t2}: change z_{t2} by assigning customer ww to the table (m+1), which increase the P by tmp1
z''_{t2}: change z_{t2} by assigning customer ww to the table (m+2), which increase the P by tmp2

assign customer ww to table (m+argmax{tmp1,tmp2})
End

(b) Iteration(2-means):

While no more changes of table assignment and dish assignment can increase P:
b=rand([0,1])
Switch (ceil(b*4)):
case 1: Randomly pick a customer from table (m+1), assign it to table (m+2) if the change increase P
case 2: Randomly pick a customer from table (m+2), assign it to table (m+1) if the change increase P
case 3: pick table (m+1), assign it the dish which increase P mostly
case 4: pick table (m+2), assign it the dish which increase P mostly
End

(4) Go back to Step I)

(III) Merge table:

(1) Randomly pick a Restaurant R, which has m tables.
(2) While no more changes of table assignment and dish assignment can increase P:
b=rand([0,1])
Switch (ceil(b*2)):
case 1: Randomly pick a table in R, merge it to the table in R which increase P mostly
case 2: Randomly pick a table in R, assign it the dish which increase P mostly
End

(3) Go back to Step I)

(IV) Split dishes:

(1) Randomly pick a Restaurant dish k, which has m tables.
(2) if(m≠1) Do 2-means++ (current: K dishes,dish assignment:z_{k0})
(a) Initialization:

Randomly pick a table t1 in from dish k to form a new dish (K+1)⇒ z_{k1}

For ww=randperm(tables from dish k except t1)
$z^{'}_{k1}$: change z_{k1} by assigning table ww to the new dish (K+1)
weight(ww)=max$\{log(p(x|\lambda, z_{k1})) - log(p(x|\lambda, z^{'}_{k1})), 0\}$ (more decrease,more weight)
End

Randomly sample another table t2 from dish k according to weight,form a new dish (K+2):z_{k2}

For ww=randperm(tables from dish k except t1,t2)
$z^{'}_{k2}$: change z_{k2} by assigning table ww to dish (K+1), which increase the P by tmp1

z''_{k2}: change z_{k2} by assigning table ww to dish (K+2), which increase the P by tmp2
assign table ww to dish (K+argmax{tmp1,tmp2})
End

(b) Iteration(2-means):

While no more changes of dish assignment can increase P:
b=rand([0,1])
Switch (ceil(b*2)):
case 1: Randomly pick a table from dish (K+1), assign it to dish (K+2) if the change increase P
case 2: Randomly pick a table from dish (K+2), assign it to dish (K+1) if the change increase P
End

(3) Go back to Step I)

(V) Merge dishes:

(1) if(K$\neq$1) Do:

While no more changes of dish assignment can increase P:
Randomly pick a dish, merge it to the dish which increase P mostly
End

(2) Go back to Step I)

<u>END</u>