Mini-Report:Decompose Restaurant

Donglai Wei

2010.6.16

1) Algorithm

GOAL: Maximize log Probability P=$logp(x, z|\lambda, \alpha, \gamma)$

(i) Make Restaurant j into one table t1 where customers following uniform distribution

(ii) Iterate until no customers are left in this uniform table t1:

  (a) For each dish k, propose to form a new table[1*] out of t1 with dish k and calculate the weight ΔP

  (b) Sample[2*] a proposal according to the weight and make the new table

(iii) TKM[3*]

(iv) Decision[4*]

NOTE::

[1*]:One naive proposal is to assign customers with overlapped words between t1 and dish k to the new table. But actually, it is a only rough approximation of the "best" new table can be made by dish k: threshold $\frac{n^w_{..k}+\phi_0}{n_{..k}+W\phi_0} \sim \frac{n^w_{..k}}{n_{..k}}$ of the overlapped word w by $\frac{1}{W}$

[2*]:ΔP has nothing to do with Probability and different scalings lead to different functionality(variability or peaky)

[3*]:In practice, merge table may not be a good idea since it's a little too greedy while the dishes are still fledging. In Local Search k, we can also restrict the tables not to have new dishes, forcing Restaurant j to be explained by old dishes.

[4*] Accept all or Reject if decrease P

2) Experiment

i) Empirical Results

[0*]:In general, Decompose Restaurant can roughly find the bars from the darkness.(See Figure 2 right)

[1*]:Proposal with threshold works better. Simply we don't want to accumulate the noise in the dish .

[2*]:During Initialization, greedy assignment is better than sampling.

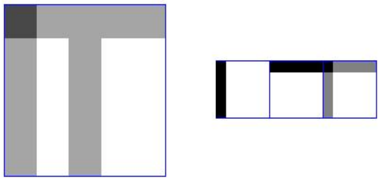

Figure 1: DR may favor mixture of bars(right) instead of pure bars(left,middle)

[3*]:Allowing new dishes in local search k works a little better.

[4*]:During Initialization, we'd better accept all new configurations which may decrease P by splitting up noisy tables.

ii) Problems(left for Decompose Dish)

1) Decompose Restaurant(DR) cannot further purify the dishes.

Since the algorithm only considers one new table at a time(a little greedy), it fails to favor pairs of dishes over one bigger dish.
Thus, if a mixture of bars happen several times, Decompose Restaurant may make it a dish even though the bars have been figured out elsewhere.

In Figure 1 are one restaurant and three dishes. Though the words in dish right has smaller words frequency, it rescues more of the uniform distributed words. Thus the Restaurant may not want to be explained by the other two dishes, even we do sampling.

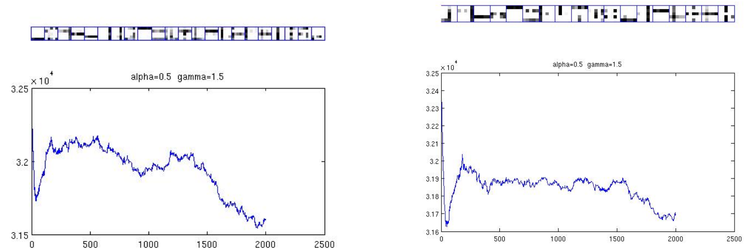

Figure 2: 1)Proprosal to make new table with threshold(right) works better

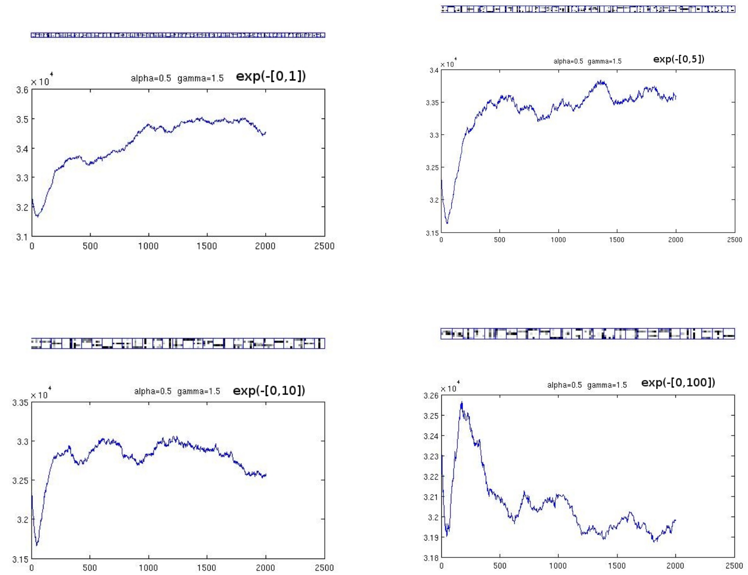

Figure 3: 2)Scale the weight to [0,r],then sample from exp(-weight),peeky works better(down-right)

iii)200 5*5 matrix

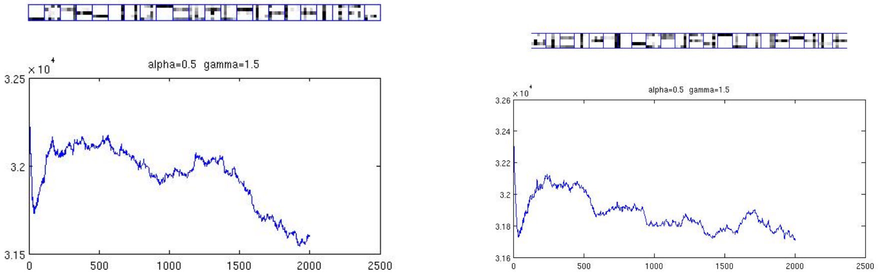

Figure 4: 3)Local Search K: allowing new dishes(left) works better than the opposite(right)

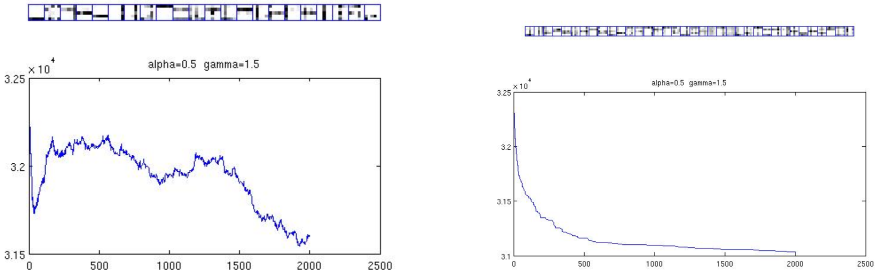

Figure 5: 4)Accept all configurations(left) works better than Accept/Reject(right)