CSE 5523: Lecture Notes 23
Gibbs sampling

Contents

23.1 Rejection sampling 1
23.2 Gibbs sampling [Geman and Geman, 1984, Carter and Kohn, 1996] 1
23.3 Example Gibbs sampling code 2
23.4 Continuous observations (Gaussian mixture model) 3

EM gets stuck in local maxima a lot.

Random exploration of the posterior distribution often gets closer to a global optimum.

23.1 Rejection sampling

We ***could*** sample the model repeatedly, then reject outcomes that don't match data:

$$\mathbf{M}_v \sim \text{Dirichlet}(\mathbf{1})$$
$$X_{n,v} \sim \text{Categorical}\left(\left(\bigotimes_{X_{n,u} \in C_{n,v}} \delta^\top_{X_{n,u}}\right)\mathbf{M}_v\right)$$

This is called **rejection sampling**.

But re-sampling just the observations is distortionary, so this would require ***many*** samples.

Solution: use conjugacy! Re-sample models using downstream (backward) distributions...

23.2 Gibbs sampling [Geman and Geman, 1984, Carter and Kohn, 1996]

Again, N training examples, each with K variables $X_{n,v}$, only some of which are observed.

(And remember C_v are conditioned-on variables, $\mathbf{f}_{v,u}$ and $\mathbf{b}_{v,w}$ are forward and backward messages.)

First, randomly initialize each random variable's model:

$$\mathbf{M}^{(0)}_v \sim \text{Dirichlet}(\mathbf{1}^{(\prod_{X_u \in C_v} |X_u|) \times |X_v|})$$

Then, for each iteration i, obtain all backward messages $\mathbf{b}^{(i)}_{n,w,v}$.

Then, going from front to back, obtain samples of each random variable:

$$X^{(i)}_{n,v} \sim \text{Categorical}\left(\left(\bigotimes_{X_{n,u} \in C_{n,v}} \delta^\top_{X_{n,u}}\right)\mathbf{M}^{(i-1)}_v \bigodot_{w \in C_v} \text{diag}(\mathbf{b}^{(i)}_{n,w,v})\right)$$

Then resample each $\mathbf{M}_v^{(i)}$ based on these variable samples:

$$\mathbf{M}_v^{(i)} \sim \text{Dirichlet}\left(\sum_n \left(\bigotimes_{X_{n,u} \in C_{n,v}} \delta_{X_{n,u}^{(i)}} \right) \left(\delta_{X_{n,v}^{(i)}} \right)^\top \right)$$

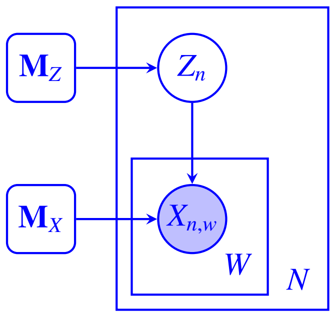
(Resampling a matrix means independently resampling each of its rows.)

This is called (an inverted variant of) **forward filtering backward sampling**.

23.3 Example Gibbs sampling code

Here's example code where one of K 'topics' is chosen for each of N W-word documents.

This fits parameters and hidden variable values for the following plate diagram:



(NOTE: here each backward message $\mathbf{b}_{n,X,Z}$ is a *product* of backward messages from X's to Z.)

```
import sys
import numpy as np
import pandas as pd

X = pd.read_csv( sys.argv[1], sep=' ' )                             ## read data
N = len(X)                                                          ## number of documents
W = len(X.columns)                                                  ## doc length in words
V = np.unique(X)                                                    ## vocab of word types
K = 2                                                               ## number of topics

M_Z = pd.DataFrame( np.random.dirichlet( np.ones( K ) ) ).T         ## initialize models
M_X = pd.DataFrame( np.random.dirichlet( np.ones( len(V) ), K ), columns=V )

xT = {}                                                             ## word Kronecker deltas
for n in range(N):                                                  ## for each document
  for w in X:                                                       ## for each word token
    xT[n,w] = pd.DataFrame( np.zeros((1,len(V))), columns=V )
    xT[n,w][ X[w][n] ] += 1                                         ## one-hots w. std cols

for i in range(3):                                                  ## for each Gibbs iter

  b_XZ = [ np.multiply.reduce( [ M_X @ xT[n,w].T for w in X ] )     ## backward messages
           for n in range(N) ]

  zT = {}                                                           ## resample variables
```

```
  for n in range(N):                                              ## for each document
    distrib = M_Z * b_XZ[n].T / (M_Z * b_XZ[n].T @ np.ones((K,K)))  ## sample topics for docs
    zT[n] = np.random.multinomial( 1, distrib.values.flatten() ).reshape((1,K))

  hparams = 1 + np.add.reduce( [ zT[n] for n in range(N) ] )        ## resample models
  M_Z = pd.DataFrame( np.random.dirichlet( hparams.flatten() ).reshape((1,K)),
                      columns=range(K) )
  hparams = 1 + np.add.reduce( [ zT[n].T @ xT[n,w] for n in range(N) for w in X ] )
  M_X = pd.DataFrame( np.stack( [ np.random.dirichlet( hparams[z] ) for z in range(K) ] ),
                      columns=V )

  print( M_Z )
  print( M_X )
```

Run on simple set of 'documents', each with three words:

```
x1 x2 x3
a b a
c b c
b a a
a b a
c b c
b a a
a b a
c b c
b a a
a b a
c b c
b a a
```

It correctly identifies word distributions for the different topics:

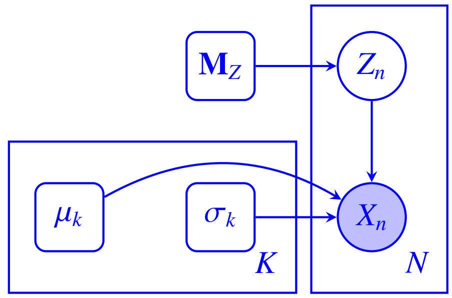
```
          0         1
0  0.527863  0.472137
          a         b         c
0  0.724047  0.248176  0.027777
1  0.003816  0.208261  0.787923

          0         1
0  0.854385  0.145615
          a         b         c
0  0.682847  0.315562  0.001591
1  0.042521  0.231869  0.725610

          0         1
0  0.614476  0.385524
          a         b         c
0  0.610351  0.335661  0.053988
1  0.058003  0.257918  0.684078
```

23.4 Continuous observations (Gaussian mixture model)

Gibbs sampling can also model continuous downstream observations (e.g. mixtures of Gaussians):



Here each observation X_n is drawn from a mixture Z_n of K different Gaussian components.

In this case the backward message still contains a likelihood of child values for each parent value:

$$(\mathbf{b}_{X_n,Z_n})_{[k]} = \mathcal{N}_{\mu_k,\sigma_k}(x_n)$$

and Gaussian parameters are sampled from the NormalGamma prior:

$$\mu_k, \sigma_k \sim \text{NormalGamma}\Bigg(\underbrace{\sum_n [\![Z_n{=}k]\!] x_n}_{\text{empirical mean of } k\text{'s}},\ N,\ \tfrac{N}{2},\ \underbrace{\sum_n [\![Z_n{=}k]\!] \Big(x_n - \underbrace{\sum_n [\![Z_n{=}k]\!] x_n}_{\text{empirical mean of } k\text{'s}} \Big)^2}_{\text{empirical variance of } k\text{'s}} \Bigg)$$

(This assumes 'uninformative' hyperparameters $\mu_0, \lambda, \alpha, \beta = 0$.)

References

[Carter and Kohn, 1996] Carter, C. K. and Kohn, R. (1996). Markov Chain Monte Carlo in Conditionally Gaussian State Space Models. *Biometrika*, 83(3):589–601.

[Geman and Geman, 1984] Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.