Collapsed Gibbs Sampler for Dirichlet Process Gaussian Mixture Models (DPGMM)

Rajarshi Das
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
rajarshd@cs.cmu.edu

Sunday 2nd November, 2014

1 Alternative definition of Dirichlet Process

We will work with the following definition of Dirichlet Process.

$$\begin{aligned} y_i|c_i, \{\mu_k, \Sigma_k\} &\sim N(\mu_{c_i}, \Sigma_{c_i}) \\ c_i|\pi &\sim Discrete(\pi_1, ..., \pi_k) \\ \{\mu_k, \Sigma_k\} &\sim NIW(\beta) \\ \pi &\sim Dirichlet(\alpha/K, ..., \alpha/K) \end{aligned}$$

A dirichlet process can be obtained by taking the limit as $K \to \infty$. [1] π defines a infinite dimensional categorical distribution with a symmetric Dirichlet prior. Also $\{\mu_k, \Sigma_k\}$, (representing the mean and the covariances of each Gaussian), have a Normal Inverse Wishart distribution as prior parameterized by $\beta = \{\mu_0, \lambda_0, \nu_0, S\}$

1.1 Quick summary of Prior distributions

1.1.1 Dirichlet distribution

Let $\pi = \{\pi_1, \pi_2, , \ldots, \pi_K\}$ be a vector of reals such that $\forall k, \pi_k \geq 0$ and $\sum_k \pi_k = 1$. Let $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ be a vector of reals with $\alpha_k > 0, \forall k$. A Dirichlet

[1] Section 2 of [2] gives a nice proof!.

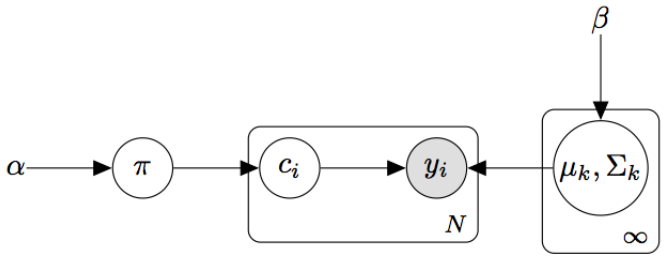

Figure 1: Dirichlet Process Gaussian Mixture Model

distribution with parameter α has a probability density function defined by

$$P(\pi|\alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \pi_k^{(\alpha_k - 1)} \tag{1}$$

where,

$$B(\alpha) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)} \tag{2}$$

Here are some facts about why a Dirichlet distribution is interesting. Below we assume, the α vector has K dimensions.

- A sample from a Dirichlet distribution is a probability vector. In other words, a Dirichlet distribution is a probability distributions over all possible categorical distributions with K dimensions.
- Dirichlet distribution is also a conjugate prior of the categorical distribution. A very important fact which we are going to utilize while deriving the collapsed Gibbs sampler.
- In a DP mixture model, the dirichlet prior has $K \to \infty$ dimensions. That means the probability vector π also has infinite components which allows data to be generated from possibly infinite number of distributions (Gaussians in this case).

1.1.2 Normal Inverse Wishart distribution

A Normal Inverse Wishart distribution is a multivariate distribution with 4 parameters $(\mu_0, \lambda_0, S_0, \nu_0)$. If

$$\mu|\mu_0, \lambda_0, \Sigma \sim N(\mu|\mu_0, \frac{1}{\lambda_0}\Sigma) \quad (3)$$

$$\Sigma|S_0, \nu_0 \sim W^{-1}(\Sigma|S_0, \nu_0) \quad (4)$$

Then (μ, Σ) has a Normal Inverse Wishart distribution with probability density function defined by

$$P(\mu, \Sigma|\mu_0, \lambda_0, S_0, \nu_0) = N(\mu|\mu_0, \frac{1}{\lambda_0}\Sigma)W^{-1}(\Sigma|S_0, \nu_0) \quad (5)$$

$$= \frac{1}{Z(D, \lambda_0, S_0, \nu_0)}|\Sigma|^{-\frac{\nu_0 + D + 2}{2}} \exp\left(-\frac{\lambda_0}{2}(\mu - \mu_0)^T\Sigma^{-1}(\mu - \mu_0) - \frac{1}{2}tr(\Sigma^{-1}S_0)\right) \quad (6)$$

where the normalization constant $Z(D, \lambda_0, S_0, \nu_0) =$

$$2^{\frac{(\nu_0+1)D}{2}}\pi^{\frac{D(D+1)}{4}}{\lambda_0}^{\frac{-D}{2}}|S_0|^{\frac{-\nu_0}{2}}\prod_{i=1}^{D}\Gamma\left(\frac{\nu_0 + 1 - i}{2}\right) \quad (7)$$

Here D denotes the dimensionality of the mean vector.
Some points worth noting.

- A sample from a Normal Inverse Wishart distribution gives us a mean and covariance matrix (which can define a multivariate gaussian distribution). We first sample a matrix from an inverse Wishart distribution parameterized by S, ν_0 and then sample a mean vector from a normal distribution parameterized by μ_0, λ_0, Σ.
- Normal Inverse Wishart distribution is a conjugate prior of a multivariate normal distribution.

2 Notation

A bit on the notation.

- N denotes the total number of data vectors
- D denotes the dimensionality of vectors

- $y_i \in \mathbb{R}^D$ denotes the i^{th} data vector.
- $y = \{y_1, y_2, y_3, \ldots, y_N\}$ (set of all data points or customers)
- $y_{-i} = \{y_j | j \neq i\}$ (set of all data points or customers except y_i)
- c_i denotes the parameter assignment of data point y_i. Those who are familiar with the Chinese Restaurant Process notation, this refers to the table assignment for customer y_i.
- c_{-i} = $\{c_j | j \neq i\}$
- $\{y_{i,k}\}$ = $\{y_j | c_j = k\}$
- $\{y_{-i,k}\}$ = $\{y_j | c_j = k, j \neq i\}$
- $N_{i,k}$ = $||\{y_{i,k}\}||$ (Size of set $y_{i,k}$)
- $N_{-i,k}$ = $||\{y_{-i,k}\}||$ (Size of set $y_{-i,k}$)

3 Collapsed Gibbs Sampling

Since we have chosen conjugate prior distributions for our parameters, so we can integrate out the parameters. This means we have to sample for less number of parameters in each round of Gibbs sweep. This technique is called collapsed Gibbs sampling or Rao-Blackwellization.
On each step of Gibbs sampling, we sample wrt the following equation

$$p(c_i = k | c_{-i}, y, \alpha, \beta) \propto p(c_i = k | c_{-i}, \alpha) p(y | c_{-i}, c_i = k, \beta) \tag{8}$$

Notice that the parameters $\{\mu_k, \Sigma_k, \pi_k\}$ does not appear in the above equation.

3.1 Prior

The prior $p(c_i = k | c_{-i}, \alpha)$ is well defined. By definition of CRP,

$$p(c_i = k | c_{-i}, \alpha) = \begin{cases} \frac{N_{-i,k}}{N+\alpha-1} & \text{if } k \text{ has been seen before (customer sits at an already occupied table)} \\ \frac{\alpha}{N+\alpha-1} & \text{if } k \text{ is new (customer sits at a new table)} \end{cases} \tag{9}$$

3.2 Likelihood

Lets consider the likelihood part $p(y|c_{-i}, c_i = k, \beta)$. Now when a new customer comes in and sits at a table, then the new likelihood (L_{new}) of data (with the new customer) can be calculated from the old likelihood (L_{old}) as

$$\begin{aligned} L_{new} &= L_{old} * \frac{p(\{y_{i,k}\}|c_{-i}, c_i = k, \beta)}{p(\{y_{-i,k}\}|c_{-i}, c_i = k, \beta)} \\ &= L_{old} * \frac{p(\{y_{i,k}\}|\beta)}{p(\{y_{-i,k}\}|\beta)} \text{(Since } \{y_{i,k}\} \text{ implies that we consider } \{c_i|c_i = k\}) \end{aligned}$$

Since L_{old} is independent of the assignment to customer k, we really need to compute it. Therefore we only need to compute $\frac{p(y_{i,k}|\beta)}{p(y_{-i,k}|\beta)}$. Now lets try to find an expression for $p(y_{i,k}|\beta)$

$$\begin{aligned} p(\{y_{i,k}\}|\beta) &= \int_{\mu}\int_{\Sigma} p(\{y_{i,k}\}, \mu, \Sigma|\beta) d\mu d\Sigma \\ &= \int_{\mu}\int_{\Sigma} p(\{y_{i,k}\}|\mu, \Sigma, \beta) p(\mu, \Sigma|\beta) \\ &= \int_{\mu}\int_{\Sigma} \prod_{i|c_i=k} p(y_i|\mu, \Sigma) p(\mu, \Sigma|\beta) \end{aligned}$$

Now,

$$\begin{aligned} p(y_i|\mu, \Sigma) &= N(y_i|\mu, \Sigma) \\ p(\mu, \Sigma|\beta) &= NIW(\mu, \Sigma|\beta) \end{aligned}$$

Since Normal and Normal Inverse Wishart distributions, form a conjugate pair, therefore $\prod_{i|c_i=k} p(y_i|\mu, \Sigma) p(\mu, \Sigma|\beta)$ also follows a Normal Inverse Wishart distribution with new parameters as given by $(\mu_N, \lambda_N, S_n, \nu_N)$ [1] where

$$\mu_N = \frac{\lambda_0 \mu_0 + N\bar{y}}{\lambda_N} \tag{10}$$

$$\begin{aligned} \lambda_N &= \lambda_0 + N \\ \nu_N &= \nu_0 + N \end{aligned}$$

$$S_N = S_0 + C + \frac{\lambda_0 N}{\lambda_N}(\bar{y} - \mu_0)(\bar{y} - \mu_0)^T \tag{11}$$

$$C = \sum_{i=1}^{N}(y_i - \bar{y})(y_i - \bar{y})^T \tag{12}$$

Important: Here N is equal to the number of customers sitting at a table (after customer y_i sat on the table). Readers shouldn't confuse it with the total number of data points.

Therefore,

$$\begin{aligned} p(y_{i,k}|\beta) &= (2\pi)^{\frac{-ND}{2}} \frac{Z(D, \lambda_N, S_N, \nu_N)}{Z(D, \lambda_0, S_0, \nu_0)} \int_{\mu} \int_{\Sigma} NIW(\mu, \Sigma|\mu_N, \lambda_N, S_N, \nu_N) d\mu d\Sigma \\ &= (2\pi)^{\frac{-ND}{2}} \frac{Z(D, \lambda_N, S_N, \nu_N)}{Z(D, \lambda_0, S_0, \nu_0)} \end{aligned}$$

where $Z(D, \lambda_0, S_0, \nu_0)$ is given by (7).
Therefore

$$\begin{aligned} \frac{p(y_{i,k}|\beta)}{p(y_{-i,k}|\beta)} &= (2\pi)^{\frac{-D}{2}} \frac{Z(D, \lambda_N, S_N, \nu_N)}{Z(D, \lambda_{N-1}, S_{N-1}, \nu_{N-1})} \\ &= (\pi)^{\frac{-D}{2}} \left(\frac{\lambda_N}{\lambda_{N-1}}\right)^{\frac{-D}{2}} \frac{|S_N|^{\frac{-\nu_N}{2}}}{|S_{N-1}|^{\frac{-\nu_{N-1}}{2}}} \prod_{i=1}^{D} \frac{\Gamma\left(\frac{\nu_0+N+1-i}{2}\right)}{\Gamma\left(\frac{\nu_0+N-i}{2}\right)} \end{aligned}$$

$$= (\pi)^{\frac{-D}{2}} \left(\frac{\lambda_N}{\lambda_{N-1}}\right)^{\frac{-D}{2}} \frac{|S_N|^{\frac{-\nu_N}{2}}}{|S_{N-1}|^{\frac{-\nu_{N-1}}{2}}} \frac{\Gamma\left(\frac{\nu_0+N}{2}\right)}{\Gamma\left(\frac{\nu_0+N-D}{2}\right)} \tag{13}$$

The term $\frac{p(y_{i,k}|\beta)}{p(y_{-i,k}|\beta)} = p(y_i|y_{-i,k}, \beta)$ is also called posterior predictive.[2] It turns out, it is equal to the density of a multivariate t-distribution with the following parameters (equation 232 of [1])

$$p(y_i|y, \beta) = t_{v_{N-1}-D+1}\left(y_i|\mu_{N-1}, \frac{S_{N-1}(k_{N-1}+1)}{k_{N-1}(v_{N-1}-D+1)}\right) \tag{14}$$

[2]Posterior predictive refers to the the prob. of seeing a new point given the posterior model.

3.3 Psuedocode

Algorithm 1 Collapsed Gibbs sampler for DPGMM

Initialize c_i for every data vector y_i to a random table.
for $iter = 1$ to T **do**
  for $i = 1$ to N **do**
    Remove the old table assignment for y_i and update the parameters accordingly of the table it got removed from.
    If any table is empty, remove and decrease K
    for $k = 1$ to K **do**
      Calculate $p(c_i = k|c_{-i}, \alpha) = \frac{N_{-i,k}}{N+\alpha-1}$
      Calculate $p(y_{i,k}|y_{-i,k}, beta)$ according to equation (14)
      Calculate $p(c_i = k|c_{-i}, y, \alpha, \beta) \propto p(c_i = k|c_{-i}, \alpha)p(y_{i,k}|y_{-i,k}, \beta)$
    end for
    {Now calculate the prior and likelihood for a new table}
    Calculate $p(c_i = k^*|c_{-i}, \alpha) = \frac{\alpha}{N+\alpha-1}$
    Calculate $p(y_{i,k^*}|\beta)$ according to equation (14) (put $N = 1$)
    Calculate $p(c_i = k^*|c_{-i}, y, \alpha, \beta) \propto p(c_i = k^*|c_{-i}, \alpha)p(y_{i,k^*}|\beta)$
    Sample a new value for c_i from $p(c_i|c_{-i}, y, \alpha, \beta)$ after normalizing.
    Update according to the new value of c_i sampled. If a new table is started, update $K = K + 1$.
  end for
end for

4 Reducing the sampling time

In this section, I will note down some optimizations which can reduce the sampling time considerably.

4.1 Maintaining Squared Sum of Customers

Whenever a customer is removed from (or added to) a table, we have to recompute the table parameters. That means re-computing μ_N and S_N by equation (10) and (11).
Now μ_N can be computed efficiently by maintaining the sum of customer vectors. So whenever a customer is removed (or added), we effectively subtract (or add) its

vector and recompute the new mean according to equation (10). Now for computing S_N, we have to compute the matrix C, which requires to go over each customer in the table. This term needs to be computed since upon removal or addition of a customer, $\bar{y}$ changes. Computing this term everytime when a customer is removed or added, could be computationally expensive. However there is another way to write (11) which is more efficient to compute. By Equation (11),

$$\begin{aligned}
S_N &= S_0 + C + \frac{\lambda_0 N}{\lambda_0 + N}(\bar{y} - \mu_0)(\bar{y} - \mu_0)^T \\
&= S_0 + \sum_{i=1}^{N}(y_i - \bar{y})(y_i - \bar{y})^T + \frac{\lambda_0 N}{\lambda_0 + N}(\bar{y} - \mu_0)(\bar{y} - \mu_0)^T \\
&= S_0 + \sum_{i=1}^{N}\left(y_i y_i^T - y_i\bar{y}^T - \bar{y}y_i^T + \bar{y}\bar{y}^T\right) + \frac{\lambda_0 N}{\lambda_0 + N}(\bar{y} - \mu_0)(\bar{y} - \mu_0)^T \\
&= S_0 + \sum_{i=1}^{N} y_i y_i^T - \sum_{i=1}^{N} y_i\bar{y}^T - \bar{y}\sum_{i=1}^{N} y_i^T + N\bar{y}\bar{y}^T + \frac{\lambda_0 N}{\lambda_0 + N}(\bar{y} - \mu_0)(\bar{y} - \mu_0)^T \\
&= S_0 + \sum_{i=1}^{N} y_i y_i^T - 2N\bar{y}\bar{y}^T + N\bar{y}\bar{y}^T + \frac{\lambda_0 N}{\lambda_0 + N}(\bar{y} - \mu_0)(\bar{y} - \mu_0)^T \\
&= S_0 + \sum_{i=1}^{N} y_i y_i^T - N\bar{y}\bar{y}^T + \frac{\lambda_0 N}{\lambda_0 + N}(\bar{y}\bar{y}^T - \mu_0\bar{y}^T - \bar{y}\mu_0^T + \mu_0\mu_0^T) \\
&= S_0 + \sum_{i=1}^{N} y_i y_i^T - \frac{1}{\lambda_0 + N}\left(N^2\bar{y}\bar{y}^T + \lambda_0 N\mu_0\bar{y}^T + \lambda_0 N\bar{y}\mu_0^T + \lambda_0^2\mu_0\mu_0^T\right) + \frac{\lambda_0 N\mu_0\mu_0^T + \lambda_0^2\mu_0\mu_0^T}{\lambda_0 + N} \\
&= S_0 + \sum_{i=1}^{N} y_i y_i^T - \frac{\left(\lambda_0\mu_0 + N\bar{y}\right)\left(\lambda_0\mu_0 + N\bar{y}\right)^T}{\lambda_0 + N} + \lambda_0\mu_0\mu_0^T \\
&= S_0 + \sum_{i=1}^{N} y_i y_i^T - \left(\lambda_0 + N\right)\mu_N\mu_N^T + \lambda_0\mu_0\mu_0^T \qquad (15)
\end{aligned}$$

This means if we maintain the squared sum of customer vectors $\left(\sum_{i=1}^{N} y_i y_i^T\right)$ then whenever a customer is removed (or added), we just have to subtract (or add) $y_i y_i^T$. Thus for each table, we have to maintain the sum of customer vectors (for μ_N) and squared sum of customer vectors (for S_N).

5 Further optimization via Cholesky decomposition if the prior covariance is positive definite

5.1 Definition

The Cholesky decomposition of a symmetric positive definite matrix A is it's decomposition into the product of a lower triangular matrix L and its transpose [3].

$$A = LL^T$$

If the matrix has dimensionality D, the time complexity of Cholesky decomposition is $O(D^3)$. Also according to the wikipedia article, when it is applicable, the Cholesky decomposition is roughly twice as efficient as the LU decomposition for solving systems of linear equations.

5.2 Rank 1 update

A rank 1 update of matrix A by vector x is of the form

$$A' = A + xx^T$$

Now, if we have the Cholesky decomposition L of A, then the Cholesky decomposition L' of A' can be computed efficiently. I will refer the reader to the wikipedia page [3] for the algorithm. The time complexity of the update is $O(D^2)$ which is an order of magnitude saving from $O(D^3)$ if we know L, the Cholesky decomposition of A.
A similar algorithm exists for Rank1 downrate (which is of the form $A = A' - xx^T$)

5.3 Putting everything in perspective

If you are wondering why the sudden crash course in linear algebra, this is because we can apply the same trick in our sampler if the prior covariance matrix is **positive definite**[3] (Identity matrix for example.).
By equation (16) S_{n+1} can be written recursively as follows

$$\begin{aligned} S_{n+1} &= S_0 + \sum_{i=1}^{N+1} y_i y_i^T - (\lambda_0 + (N+1))\,\mu_{N+1}\mu_{N+1}^T + \lambda_0\mu_0\mu_0^T \\ &= S_n + y_{n+1}y_{n+1}^T - (\lambda_0 + (N+1))\,\mu_{N+1}\mu_{N+1}^T + (\lambda_0 + N)\,\mu_N\mu_N^T \end{aligned}$$

[3] http://en.wikipedia.org/wiki/Positive-definite_matrix

Writing μ_n interms of μ_{n+1}

$$\mu_n = \frac{(\lambda_0 + N + 1)\mu_{n+1} - y_{n+1}}{\lambda_0 + N}$$

we get,

$$\begin{aligned} S_{n+1} &= S_n + y_{n+1}y_{n+1}^T - (\lambda_0 + (N+1))\,\mu_{N+1}\mu_{N+1}^T \\ &\quad + \frac{((\lambda_0 + N + 1)\mu_{n+1} - y_{n+1})((\lambda_0 + N + 1)\mu_{n+1} - y_{n+1})^T}{(\lambda_0 + N)} \\ &= S_n + \frac{(\lambda_0 + N + 1)}{\lambda_0 + N}(\mu_{N+1} - y_{N+1})(\mu_{N+1} - y_{N+1})^T \end{aligned} \tag{16}$$

Equation (16) implies that S_{n+1} can be obtained from S_n by a Rank 1 update. Therefore if we know the cholesky decomposition of S_n, then the decomposition of S_{n+1} can be obtained in $O(D^2)$ time.

5.4 How Cholesky decomposition helps speed up sampling.

If the prior covariance is positive definite, then we could do all the steps of the sampling with just the Cholesky decomposition of the covariance matrix of each table. We dont need to store the covariance matrices required to compute the $|\Sigma|$ and Σ^{-1}, (which are both $O(D^3)$ operations).

5.4.1 Determinant

The determinant of a matrix Σ, can be computed from its cholesky decomposition L, by the following

$$log(|\Sigma|) = 2 * \sum_{i=1}^{D} log(L(i,i))$$

It is clearly an O(D) operation!

5.4.2 Inverse

Next, we need to be able to compute Σ^{-1} so that we can compute $(x-\mu)^T\Sigma^{-1}(x-\mu)$ (as required to compute the density of the multivariate t- distribution[4]). We

[4] http://en.wikipedia.org/wiki/Multivariate_t-distribution

can compute this expression, using L (the cholesky decomposition of Σ) by the following way. Representing $b = (x - \mu)$, we have to solve $b^T \Sigma^{-1} b$.

$$\begin{aligned} b^T \Sigma^{-1} b &= b^T (LL^T)^{-1} b \\ &= b^T (L^{-1})^T L^{-1} b \\ &= (L^{-1} b)^T (L^{-1} b) \end{aligned}$$

Therefore we have to compute $(L^{-1}b)$ and multiply its transpose with itself. Now $(L^{-1}b)$ is the solution of

$$Lx = b$$

Also remember L is a lower triangular matrix, therefore the above equation can be solved very efficiently using forward substitution!. Hence $(x - \mu)^T \Sigma^{-1} (x - \mu)$ can be computed in $O(D^2)$ time. Therefore we can compute everything using the choelsky decomposition L of Σ and hence we donot need to store (or compute) Σ.

6 Conclusion

In this tutorial, I have tried to explain and derive the collapsed Gibbs sampling algorithm for Dirichlet Process Mixture Models when the table distributions are multivariate gaussians with a Normal Inverse Wishart prior. I have also shown a way to speed up sampling (by an order of magnitude) by using Rank 1 cholesky updates of covriance matrces[5]. This is ofcourse possible when the prior covariance matrix is a positive definite matrix.

7 Acknowledgments

Many thanks to my colleague and friend in CMU, Manzil Zaheer (manzil@zaheer.ml), for introducing me to the concepts of Cholesky decomposition and proving that the covariance update is indeed a Rank 1 update!.

[5]This technique, in my opinion is quite new to the NLP research community

References

[1] Kevin P. Murphy. Conjugate bayesian analysis of the gaussian distribution. Technical report, UBC, 2007.

[2] Radford M. Neal. Markov chain sampling methods for dirichlet process mixture models. *JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS*, 9(2):249–265, 2000.

[3] Wikipedia. Cholesky decomposition. [Online; accessed 01-October-2014].