Fast Collapsed Gibbs Sampler for Dirichlet Process Gaussian Mixture Models using Rank 1 Cholesky updates

Rajarshi Das

rajarshd@cs.cmu.edu

http://www.cs.cmu.edu/~rajarshd/

c-lab lecture

7th October 2014

template adapted from
https://www.sharelatex.com/templates/presentations/radboud-university-beamer-(version-1)

Overview

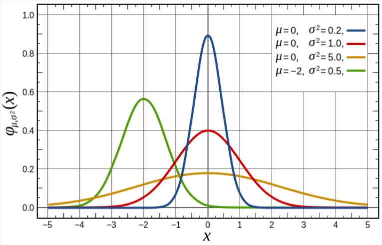
1. Distributions of interest
2. Dirichlet Process
3. Finite, Infinite and Dirichlet Process Mixture Models
4. Dirichlet Process <u>Gaussian</u> Mixture Models
5. Gibbs sampler for DPGMM
6. Linear Algebra Crash Course
7. Fast Gibbs Sampler for DPGMM
8. Results

Overview

Gaussian Distribution

- 2 parameter distribution (μ and σ)

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



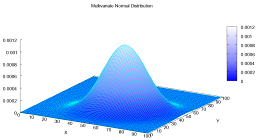
[1]

[1] images from https://en.wikipedia.org/wiki/Normal_distribution

Multivariate Gaussian Distribution

- Multivariate generalization of Gaussian distribution
- μ is a vector and σ becomes covariance matrix Σ

$$f_{\mathbf{x}}(x_1, \ldots, x_k) = \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$



Normal Inverse Wishart

- 4 parameter family $(\mu_0, \kappa_0, \Sigma_0, \nu_0)$

$$\begin{aligned}
\Sigma|\Sigma_0, \nu_0 &\sim W^{-1}(\Sigma|\Sigma_0, \nu_0) \\
\mu|\mu_0, \kappa_0, \Sigma &\sim N(\mu|\mu_0, \frac{1}{\kappa_0}\Sigma) \\
(\mu, \Sigma) &\sim N(\mu|\mu_0, \frac{1}{\kappa_0}\Sigma)W^{-1}(\Sigma|\Sigma_0, \nu_0)
\end{aligned}$$

Multivariate t

- Multivariate generalization of student's *t*-distribution
- 3 paramters (μ, Σ, ν)

$$\frac{\Gamma\left[(\nu+p)/2\right]}{\Gamma(\nu/2)\nu^{p/2}\pi^{p/2}\left|\boldsymbol{\Sigma}\right|^{1/2}\left[1+\frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]^{(\nu+p)/2}}$$

Dirichlet Distribution

- A dirichlet distribution is a probability distribution over categorical distributions, parameterized by a vector α of fixed length K.
- $X =< x_1, x_2, ..., x_K >, x_i \in [0, 1], \sum_i x_i = 1$,
  is dirichlet distributed with parameter α if,

Dirichlet Distribution

- A dirichlet distribution is a probability distribution over categorical distributions, parameterized by a vector α of fixed length K.
- $X =< x_1, x_2, ..., x_K >, x_i \in [0, 1], \sum_i x_i = 1$, is dirichlet distributed with parameter α if,

pdf

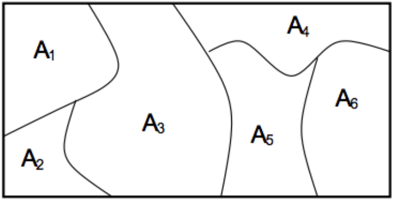
$$p(X|\alpha) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_{k=1}^{K} \Gamma\left(\alpha_k\right)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

Overview

Dirichlet Process

- A Dirichlet Process (DP) is also a distribution over discrete distributions.
  - This time with no fixed dimensions. In other words the 'K' is not fixed (could be infinite!)
- A DP has 2 parameters
  1. base distribution G_0
  2. concentration parameter α

Definition



[2]

$$G \sim DP(G_0, \alpha)$$

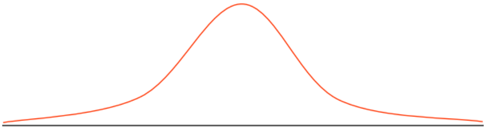
if for any partition $(A_1, ..., A_K)$ of a measurable space X

$$(G(A_1), ..., G(A_K)) \sim Dirichlet(\alpha G_0(A_1), ..., \alpha G_0(A_K))$$

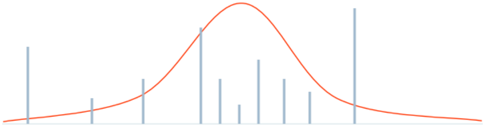
[2] Image from `http://www.columbia.edu/~jwp2128/Teaching/E6892/papers/mlss2007.pdf`

Example (slide adapted from https://www.cs.cmu.edu/~kbe/dp_tutorial.pdf)

- For example, if the base distribution G_0 is a Gaussian.



- The sampled distribution $G \sim DP(G_0, \alpha)$ would look like



Example

Realizations From the Dirichlet Process Look Like a Used Dartboard.[3]

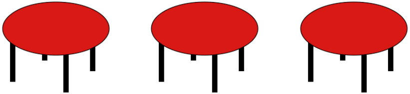
[3] https://www.ee.washington.edu/techsite/papers/documents/UWEETR-2010-0006.pdf

Posterior Predictive distribution of DP

$$
\begin{aligned}
G &\sim DP(G_0, \alpha) \\
\theta_1, ..., \theta_n &\sim G \\
\theta_{n+1} | \theta_1, ..., \theta_n &\sim \frac{1}{(\alpha + n)} (\alpha G_0 + \Sigma_{i=1}^{n} \delta_{\theta_i})
\end{aligned}
$$

- Also known as Blackwell-MacQueen Urn Scheme
- Has the rich getting richer property
- Chinese Restaurant Process!

Chinese Restaurant Process

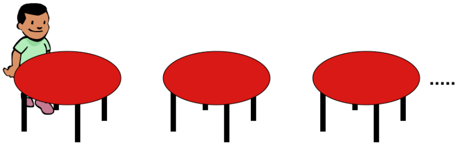


$$\mathrm{P}(c_i = c|\mathbf{c}_{-i}) =$$

[4]

[4] Graphic from http://nlp.stanford.edu/~grenager/papers/dp_2005_02_24.ppt

Chinese Restaurant Process

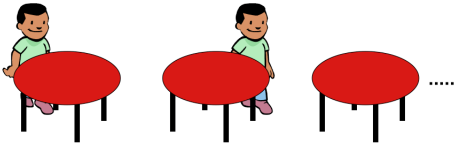


$P(c_i = c|\mathbf{c}_{-i}) =$ 1 0 0

[5]

[5] Graphic from
http://nlp.stanford.edu/~grenager/papers/dp_2005_02_24.ppt

Chinese Restaurant Process

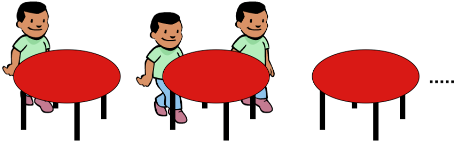


$P(c_i = c \mid \mathbf{c}_{-i}) =$	1	0	0
	$\frac{1}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$	0

[6]

[6] Graphic from
http://nlp.stanford.edu/~grenager/papers/dp_2005_02_24.ppt

Chinese Restaurant Process

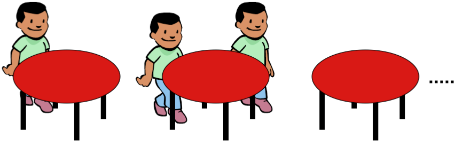


$\mathrm{P}(c_i = c \mid \mathbf{c}_{-i}) =$	1	0	0
	$\frac{1}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$	0
	$\frac{1}{2+\alpha}$	$\frac{1}{2+\alpha}$	$\frac{\alpha}{2+\alpha}$

[7]

[7] Graphic from
`http://nlp.stanford.edu/~grenager/papers/dp_2005_02_24.ppt`

Chinese Restaurant Process



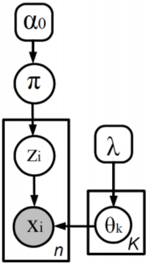
$P(c_i = c \mid \mathbf{c}_{-i}) =$	1	0	0
	$\frac{1}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$	0
	$\frac{1}{2+\alpha}$	$\frac{1}{2+\alpha}$	$\frac{\alpha}{2+\alpha}$
	$\frac{1}{3+\alpha}$	$\frac{2}{3+\alpha}$	$\frac{\alpha}{3+\alpha}$ [8]

[8] Graphic from
`http://nlp.stanford.edu/~grenager/papers/dp_2005_02_24.ppt`

Overview

Finite Mixture Models (FMM)

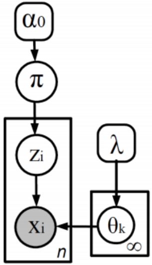
- In FMM, we assume data is generated from K distributions.



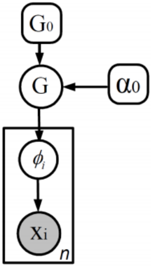
$$
\begin{aligned}
\forall k, \theta_k | \lambda &\sim G_0(\lambda) \\
\pi | \alpha_0 &\sim Dir(\alpha_0/K, \alpha_0/K, ..., \alpha_0/K) \\
z_i | \pi &\sim Discrete(\pi) \\
x_i | z_i, \{\theta_k\} &\sim F(\theta_{z_i})
\end{aligned}
$$

Infinite Mixture Models (IMM)

- In Infinite mixture models, we don't know how many mixtures generated the data, so we donot fix a value of K and let the data decide!



Dirichlet Process Mixture Models (DPMM)



$$
\begin{aligned}
G &\sim DP(G_0, \alpha) \\
\forall i, \phi_i &\sim G \\
\forall i, x_i &\sim F(\phi_i)
\end{aligned}
$$

DPMM and IMM are the same!

- Turns out that IMM are same as DPGMM when $K \to \infty$

$$P(z_i = k | z_{-i}) = \begin{cases} \frac{n_{i,k} + \frac{\alpha}{K}}{n - 1 + \alpha} & \text{if } k \text{ is seen before} \\ \frac{\alpha}{n - 1 + \alpha} & \text{if } k \text{ is new} \end{cases}$$

Overview

DPGaussianMM

- In DPGMM, each table is endowed with a Gaussian distribution.

$$
\begin{aligned}
G &\sim DP(G_0, \alpha) \\
\forall i, \phi_i &\sim G \\
\forall i, x_i &\sim F(\phi_i)
\end{aligned}
$$

- F is Gaussian distribution.
- if G_0 is a conjugate distribution to F, then inference via Gibbs sampling becomes easy[a].
  - if covariance is fixed, then G_0 is Normal
  - else if both mean and covariance are unknown, then G_0 is Normal Inverse Gamma (univariate) or Normal Inverse Wishart (multivariate)

[a] Algorithm 1,2,3 of Neal(2000)

Overview

Posterior Inference for DPGMM

- So I believe, that the data has been generated by a Dirichlet Process Mixture Model. Now from the observed data we want to infer the parameters of the distributions which generated the data. (a.k.a Bayesian Inference)
- What are the params of my Posterior distribution?
  - The mean and variances associated with each table.
    - we don't know a priori how many of them are there, btw!
  - The table assignment of each data point (customer).
  - p($\{\mathbf{z}\}, \{\mu, \Sigma\}|$ Data, Hyperparams) $=$ p($\{\mu, \Sigma\}|$Hyperparams) * p($\{\mathbf{z}\}|$Everything)
  - If the prior of the table params is conjugate to the likelihood, then we can integrate out $\{\mu, \Sigma\}$ – collapsing!
  - p($\{\mathbf{z}\}|$ Data, Hyperparams)

- Exact computation of posterior is intractable, however we can use MCMC techniques to sample from posterior distribution.
- Collapsed Gibbs Sampler

$$p(z_i = k | z_{-i}, X, \alpha, \lambda) \quad = \quad p(z_i = k | z_{-i}, \alpha) p(X | z, \lambda)$$

  - $p(z_i = k | z_{-i}, \alpha)$ is well defined by CRP.
  - $p(X | z, \lambda)$ is the data likelihood. Essentially reduces to computing $p(x_i | x_{-i}, z, \lambda)$
  - After some math, $p(x_i | x_{-i}, z, \lambda)$ reduces to

$$t_{v_{N-1} - D + 1} \left(x_i | \mu_{N-1}, \frac{\Sigma_{N-1}(k_{N-1} + 1)}{k_{N-1}(v_{N-1} - D + 1)} \right)$$

Gibbs sampler for Inference

- Now that we have analytical form of everything, the Gibbs sampling algorithm becomes
- Randomly initialize customer to tables. Calculate the table params. (μ and Σ)
- For each customer
  - Remove it from the current table. Update the parameters of the table.
  - for each table 1...K
    - Compute prior prob for the customer to sit in that table.
    - Compute posterior predictive likelihood to sit in the table.
    - Multiply them to get $p(z_i = k | z_{-i}, X, \alpha, \lambda)$
  - Normalize the vector of probabilities and sample for a table number. Update the parameters of the table.

Updating the table parameters

- Every table has 2 parameters - a μ and Σ matrix (which define a gaussian distribution)
- For conjugacy we have put a $\mathrm{NIW}(\mu_0, \Sigma_0, \kappa_0, \nu_0)$ prior on them.
- Since Gaussian and NIW form a conjugate prior, the posterior distribution of μ and Σ also forms a NIW distribution with the following parameters

$$
\begin{aligned}
\mu_n &= \frac{\kappa_0 \mu_0 + n\bar{x}}{\kappa_0 + n} \\
\Sigma_n &= \Sigma_0 + \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T + \frac{\kappa_0 n}{\kappa_0 + n} (\bar{x} - \mu_0)(\bar{x} - \mu_0)^T \\
\kappa_n &= \kappa_0 + n \\
\nu_n &= \nu_0 + n
\end{aligned}
$$

Runtime Analysis

- Lets look at the following code snippet from the Gibbs sampler algorithm
  - for each table $1 \ldots K$
    - Compute prior prob for the customer to sit in that table.
    - Compute posterior predictive likelihood to sit in the table. (this is a multivariate t-distribution)
  - Computing the density under t-distribution requires computing the determinant and inverse of a covariance matrix.
    - Both of these operations are very expensive. $O(D^3)$ operations, for a DXD matrix
    - Therefore the loop takes $O(KD^3)$ operations.

Runtime Analysis -II

- Randomly initialize customer to tables. Calculate the table params. (μ and Σ)
  - for μ - $O(D)$ - for each table
  - for Σ - $O(D^2)$ - for each table
  - Also for each Σ
    - Σ^{-1} and $|\Sigma|$ - $O(D^3)$
- For the number of iterations you want to run.
- For each customer
  - Remove it from the current table. Update the parameters of the table. - $O(D)$ for μ and $O(D^2)$ for Σ
  - for each table $1 \ldots K$
    - Compute prior prob for the customer to at table. - $O(1)$
    - Compute posterior predictive likelihood to sit in the table. - $O(D^2)$
    - Multiply them to get $p(z_i = k | z_{-i}, X, \alpha, \lambda)$ - $O(1)$
  - Normalize the vector of probabilities and sample for a table number. Update the parameters of the table. - $O(K) + O(D) + O(D^2) + O(D^3)$

Overview

- Cholesky decomposition: Decomposition of a symmetric positive definite matrix into the product of a lower triangular matrix and its conjugate transpose.

$$A = LL^T$$

$$\begin{pmatrix} 4 & 12 & -16 \\ 12 & 37 & -43 \\ -16 & -43 & 98 \end{pmatrix} = \begin{pmatrix} 2 & & \\ 6 & 1 & \\ -8 & 5 & 3 \end{pmatrix} \begin{pmatrix} 2 & 6 & -8 \\ & 1 & 5 \\ & & 3 \end{pmatrix}$$

- Computing the cholesky decomposition takes $O(D^3)$ time!.

Cholesky updates

- Suppose A is a positive definite matrix with L as its cholesky decomposition.
- Now if we obtain A' from A by an update of the form

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

- then the cholesky decomposition L' of A' can be obtained by an update operation on L. (Rank 1 update)
- Similarly if we have $A = A'$ - xx^T, then we can perform a Rank1 downdate to get L from L'

Cholesky updates

```
function [L] = cholupdate(L,x)
    p = length(x);
    x = x';
    for k=1:p
        r = sqrt(L(k,k)^2 + x(k)^2);
        c = r / L(k, k);
        s = x(k) / L(k, k);
        L(k, k) = r;
        L(k,k+1:p) = (L(k,k+1:p) + s*x(k+1:p)) / c;
        x(k+1:p) = c*x(k+1:p) - s*L(k, k+1:p);
    end
end
```
[9]

- This algorithm is $O(D^2)$!

[9] Image from http://en.wikipedia.org/wiki/Cholesky_decomposition

Nice properties

- $|A|$ can be computed from L by

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

- Now lets try to compute $b^T A^{-1} b$

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

- Therefore compute $(L^{-1}b)$ and multiply its transpose with itself

$(L^{-1}b)$

- $(L^{-1}b)$ is the solution of

$$Lx = b$$

- Remember L is a lower triangular matrix, therefore the above equation can be solved very efficiently using forward substitution!

$$\begin{array}{ccccccc}
l_{1,1}x_1 & & & & & = & b_1 \\
l_{2,1}x_1 & + & l_{2,2}x_2 & & & = & b_2 \\
\vdots & & \vdots & \ddots & & & \vdots \\
l_{m,1}x_1 & + & l_{m,2}x_2 & + \cdots + & l_{m,m}x_m & = & b_m
\end{array}$$

Overview

Putting back in perspective

- Recall the posterior update equation for covariance matrix

$$\begin{aligned}\Sigma_n &= \Sigma_0 + \sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T + \frac{\kappa_0 n}{\kappa_0 + n}(\bar{x} - \mu_0)(\bar{x} - \mu_0)^T \\ &= \Sigma_0 + \sum_{i=1}^{N} x_i x_i^T - (\kappa_0 + n)\mu_n \mu_n^T + \kappa_0 \mu_0 \mu_0^T\end{aligned}$$

- Now we can write the above definition recursively as

$$\begin{aligned}\Sigma_n &= \Sigma_{n-1} + x_n x_n^T - (\kappa_0 + n)\mu_n \mu_n^T + (\kappa_0 + n - 1)\mu_{n-1}\mu_{n-1}^T \\ &= \Sigma_{n-1} + \frac{\kappa_0 + n}{\kappa_0 + n - 1}(x_n - \mu_n)(x_n - \mu_n)^T\end{aligned}$$

- Looks similar to $A' = A + xx^T$?

Putting back in perspective

- Recall pdf of multi-variate t-distribution

$$\frac{\Gamma\left[(\nu+p)/2\right]}{\Gamma(\nu/2)\nu^{p/2}\pi^{p/2}\left|\boldsymbol{\Sigma}\right|^{1/2}\left[1+\frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]^{(\nu+p)/2}}$$

- If we know the cholesky decomposition of Σ, then we can compute $(x-\mu)^T\Sigma^{-1}(x-\mu)$ from L using similar trick shown before for computing $b^T L^{-1} b$

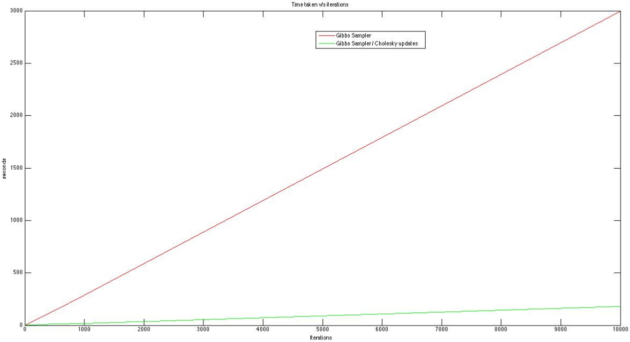
Runtime Analysis -II

- Randomly initialize customer to tables. Calculate the table params. (μ and Σ)
  - for μ - $O(D)$ - for each table
  - for Σ - (do not need to compute Σ)
  - Also for each Σ
    - Σ^{-1} and $|\Sigma|$ - $O(D^2)$
- For the number of iterations you want to run.
- For each customer
  - Remove it from the current table. Update the parameters of the table. - $O(D)$ for μ and $O(D^2)$ for Σ
  - for each table 1...K
    - Compute prior prob for the customer to at table. - $O(1)$
    - Compute posterior predictive likelihood to sit in the table. - $O(D^2)$
    - Multiply them to get $p(z_i = k | z_{-i}, X, \alpha, \lambda)$ - $O(1)$
  - Normalize the vector of probabilities and sample for a table number. Update the parameters of the table. - $O(K) + O(D)$ $+ O(D^2)$

Overview

Speed Test 1

- Just 100 word vectors each of 100 dimensions.
- Both sampler run for 10,20,50,100,500,1000,10000 iterations.



Caveats

- All of these is possible if the prior covariance Σ_0 is positive definite.
- In practice, we often tend to put Identity as Σ_0, which is positive definite.

Code

- Java Implementation available at
  `https://github.com/rajarshd/DPGMM`
- Plan to implement in C++ too.

Acknowledgements

- Thanks to my friend Manzil Zaheer for introducing me to Cholesky decomposition and proving that the covariance update was indeed a rank 1 update.