DP Mixture

Machine Learning Course

Spring 2015

Tsinghua University

Goal

- Group the given data set into several clusters
- Automatically infer the *number* of clusters
- Roughly have a sense of how nonparametric Bayesian models work

DP Mixture Model

- Data: $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{n}$
- Parameters (stochastic):
  - "membership" indicator: $\mathbf{z} = (z_1, \ldots, z_n),\ z_i \in \{1, \ldots, K\}$
  - component parameter: $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_K)$

Inferred from data, $K \leq n$

- Infer the posterior distribution of parameters

$$p(\boldsymbol{\phi}, \mathbf{z} | \mathcal{D}) \propto p_0(\boldsymbol{\phi}) p_0(\mathbf{z}) p(\mathcal{D} | \boldsymbol{\phi}, \mathbf{z})$$

To ease inference, we adopt conjugate prior

as defined in CRP

likelihood, same as in finite mixture models

DP Mixture Model (cont.)

- For likelihood $p(\mathcal{D}|\boldsymbol{\phi}, \mathbf{z})$, we assume data in each component follows a Gaussian distribution, and all components share a common covariance matrix
  - $\phi_k = \mu_k$
  - $p(\mathbf{x}_i|\boldsymbol{\phi}, \mathbf{z}) \sim \mathcal{N}(\mu_{z_i}, \Sigma)$, for now we fix $\Sigma = I$ and do not infer it.

this means we suppose that data was generated from isotropic Gaussians

- For prior $p_0(\boldsymbol{\phi})$, we use the conjugate prior
  - $p_0(\phi_k) \sim \mathcal{N}(0, \sigma^2 I)$, we can also fix $\sigma = 1$ for simplicity.

the means of isotropic Gaussians also su
Gaussian distribution

- For prior $p_0(\mathbf{z})$, we use the CRP representation of DP, which yields the following local conditional distribution
  - $p(z_i = k|\mathbf{z}_{-i}) = \dfrac{n_k}{n-1+\alpha} \quad (\text{if } k \in \{1, \ldots, K\})$
  - $p(z_i = K+1|\mathbf{z}_{-i}) = \dfrac{\alpha}{n-1+\alpha}$
  - $n_k = \sum_{j \in \{1,\ldots,n\}\backslash\{i\}} \delta_{z_j,k}$: number of *other* people at table k

current number of components

Kronecker delta

Inference by Gibbs sampling

- Randomly initialize $\mathbf{z}, \boldsymbol{\phi}$
- Sample each z_i from
  - old component ($k \in \{1, \ldots, K\}$):

$$p(z_i = k|\mathcal{D}, \boldsymbol{\phi}, \mathbf{z}_{-i}) \propto p(z_i = k|\mathbf{z}_{-i})p(\mathbf{x}_i|\mathbf{z}, \boldsymbol{\phi})$$
$$= \frac{n_k}{n-1+\alpha} \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x}_i - \mu_k)^\top \Sigma^{-1}(\mathbf{x}_i - \mu_k)\}$$

  - new component:

$$p(z_i = K+1|\mathcal{D}, \boldsymbol{\phi}, \mathbf{z}_{-i}) \propto p(z_i = K+1|\mathbf{z}_{-i})p(\mathbf{x}_i|\mathbf{z}, \boldsymbol{\phi})$$
$$= p(z_i = K+1|\mathbf{z}_{-i}) \int p_0(\phi_{K+1})p(\mathbf{x}_i|z_i, \boldsymbol{\phi}, \phi_{K+1})d\phi_{K+1}$$
$$= \frac{\alpha}{n-1+\alpha} \frac{1}{(2\pi)^{d/2}\sigma^d} \frac{|\Sigma'|^{1/2}}{|\Sigma|^{1/2}} \exp\left\{\frac{1}{2}\mathbf{x}_i^\top(\Sigma^{-1}\Sigma'\Sigma^{-1} - \Sigma^{-1})\mathbf{x}_i\right\}$$
$$(\text{where } \Sigma' = \left(\frac{1}{\sigma^2}I + \Sigma^{-1}\right)^{-1})$$

Inference by Gibbs sampling (cont.)

- Sample each ϕ_k from

$$
\begin{aligned}
p(\phi_k|\mathcal{D},\mathbf{z}) &\propto p_0(\phi_k)\prod_{i=1}^{n} p(\mathbf{x}_i|\mathbf{z},\boldsymbol{\phi}) \\
&= p_0(\phi_k)\prod_{i|z_i=k} p(\mathbf{x}_i|\mathbf{z},\boldsymbol{\phi}) \\
&\propto \exp\left\{-\frac{1}{2\sigma^2}\mu_k^\top\mu_k - \frac{1}{2}\sum_{i|z_i=k}(\mathbf{x}_i-\mu_k)^\top\Sigma^{-1}(\mathbf{x}_i-\mu_k)\right\} \\
&\sim \mathcal{N}(\mu_k', \Sigma_k') \\
&\quad (\text{where } \mu_k' = \Sigma_k'\Big(\Sigma^{-1}\sum_{i|z_i=k}\mathbf{x}_i\Big),\ \Sigma_k' = \Big(\frac{1}{\sigma^2}I + c_k\Sigma^{-1}\Big)^{-1},\ c_k = \sum_{i=1}^{n}\delta_{z_i,k})
\end{aligned}
$$

Optional: Collapsed Gibbs sampling

- Sample only in collapsed space $\mathbf{z} = (z_1, \ldots, z_n)$, with ϕ integrated out
  - old component:

$$q_{k'}(\phi_{k'}) = p_0(\phi_{k'}) \prod_{j \neq i | z_j = k'} p(\mathbf{x}_j | \boldsymbol{\phi}, \mathbf{z})$$

$$\begin{aligned}
p(z_i = k | \mathcal{D}, \mathbf{z}_{-i}) &\propto p(z_i = k | \mathbf{z}_{-i}) p(\mathcal{D} | \mathbf{z}) \\
&= p(z_i = k | \mathbf{z}_{-i}) \int p_0(\boldsymbol{\phi}) p(\mathcal{D} | \boldsymbol{\phi}, \mathbf{z}) d\boldsymbol{\phi} \\
&= p(z_i = k | \mathbf{z}_{-i}) \int p(\mathbf{x}_i | \boldsymbol{\phi}, \mathbf{z}) \prod_{k'=1}^{K} \boxed{q_{k'}(\phi_{k'})} d\boldsymbol{\phi} \\
&= p(z_i = k | \mathbf{z}_{-i}) \boxed{\int \mathcal{N}(\mathbf{x}_i | \phi_k, \Sigma) q_k(\phi_k) d\phi_k} \prod_{k' \neq k} \boxed{\int q_{k'}(\phi_{k'}) d\phi_{k'}}
\end{aligned}$$

same old tricks with Gaussian-like functions

  - new component:

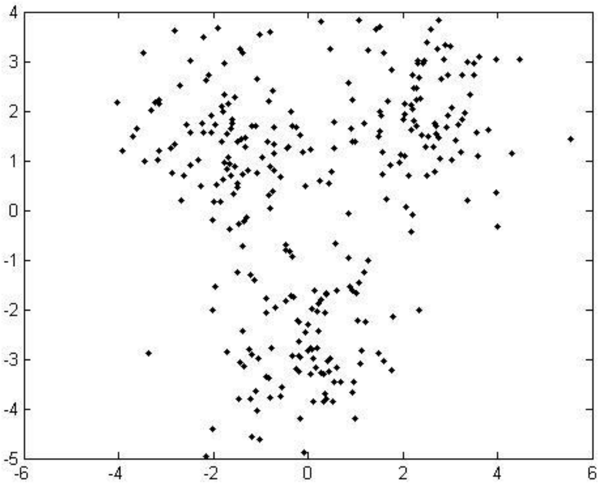
$$\begin{aligned}
p(z_i = K+1 | \mathcal{D}, \mathbf{z}_{-i}) &\propto p(z_i = K+1 | \mathbf{z}_{-i}) p(\mathcal{D} | \mathbf{z}) \\
&= p(z_i = k | \mathbf{z}_{-i}) \int p_0(\boldsymbol{\phi}) p_0(\phi_{K+1}) p(\mathcal{D} | \boldsymbol{\phi}, \phi_{K+1}, \mathbf{z}) d\boldsymbol{\phi} d\phi_{K+1} \\
&= p(z_i = k | \mathbf{z}_{-i}) \boxed{\int p_0(\phi_{K+1}) \mathcal{N}(\mathbf{x}_i | \phi_{K+1}, \Sigma) d\phi_{K+1}} \prod_{k'=1}^{K} \int q_{k'}(\phi_{k'}) d\phi_{k'}
\end{aligned}$$

You can find this term on previous slides

Generate Data

- We recommend that you use Matlab to generate the data in two and three dimensions for the ease of visualization.
- At least three different settings should be reported, e.g., with different numbers of mixture components, with zero mean or non-zero mean Guassian likelihood models to generate the data.
- Suppose the data was generated from the mixture of 3 isotropic Gaussians. A naïve way to do this in Matlab is:

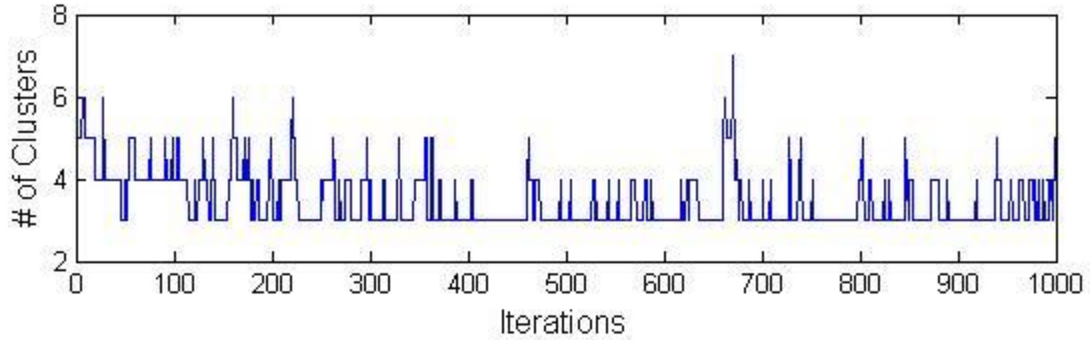
```
%Generating data
dat1=normrnd(0,1,100,2);
dat1(:,1)=dat1(:,1)+2.4;
dat1(:,2)=dat1(:,2)+2;
dat2=normrnd(0,1,100,2);
dat2(:,1)=dat2(:,1)-1.8;
dat2(:,2)=dat2(:,2)+1.4;
dat3=normrnd(0,1,100,2);
dat3(:,1)=dat3(:,1)-0.2;
dat3(:,2)=dat3(:,2)-2.6;
```



Here we fix $\Sigma = I$ for simplicity.

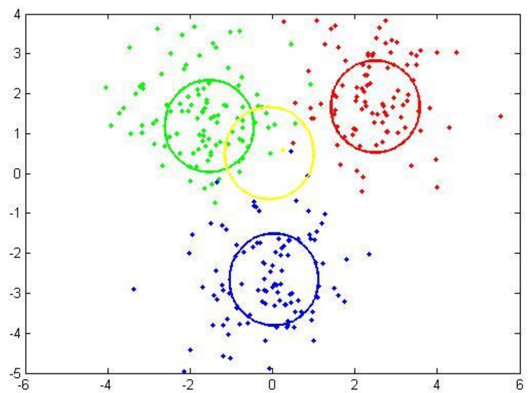
Learning

Then do Gibbs sampling as in P5-6 (and Collapsed Gibbs sampling as in P7(Optional)). You can watch how the number of clusters vary through the process of learning. You can even plot a chart like this:

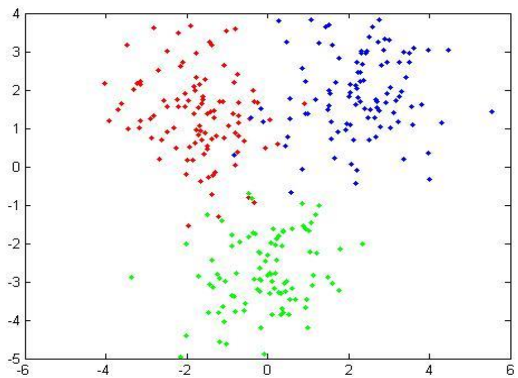


We can see how our data “decide” the number of clusters.

For Gibbs sampling, we inference z and ϕ.



For Collapsed Gibbs sampling, we only sample z.



How to evaluate

- Posterior result should be a trade-off between the following 3:
  - difference between actual and expected number of clusters a priori:

$$D(K;\alpha) = K - \mathrm{E}_{p_0}[K(n)], \text{where } \mathrm{E}_{p_0}[K(n)] = \sum_{i=1}^{n} \frac{\alpha}{i-1+\alpha} \simeq \alpha \log(1+n/\alpha)$$

  - Mahalanobis distance of all component means to their means a priori:

$$D_M(\boldsymbol{\phi};\sigma) = \frac{1}{\sigma^2} \sum_{k=1}^{K} |\phi_k^\top \phi_k|^{\frac{1}{2}}$$

  - Mahalanobis distance of all data points to their centers:

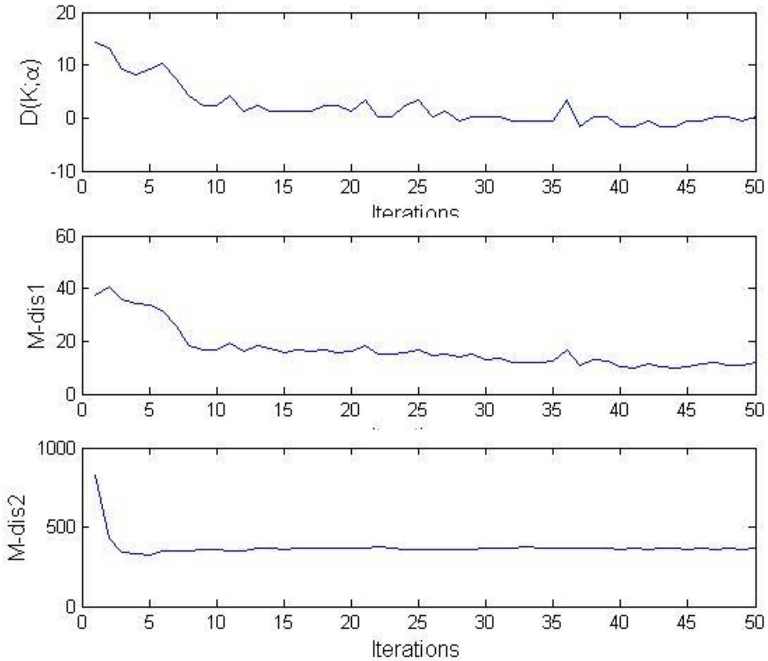
$$D_M(\mathcal{D};\mathbf{z},\boldsymbol{\phi}) = \sum_{i=1}^{n} |(\mathbf{x}_i - \mu_{z_i})^\top \Sigma^{-1} (\mathbf{x}_i - \mu_{z_i})|^{\frac{1}{2}}$$

- Actually we shall take *expectation* of the 3 estimators, but in Gibbs sampling, we approximate this by taking *sample average*

$$\text{e.g.} \quad \mathrm{E}_{p(K)}[D(K)] \approx \frac{1}{t} \sum_{s=1}^{t} D(K^{(s)})$$

How to evaluate (cont.)

- We want you to report
  - curve of the 3 estimators during the whole Gibbs sampling process
  - expected estimators (sample average) after mixing (take $t = 10$)



M-dis1= $$D_M(\boldsymbol{\phi};\sigma) = \frac{1}{\sigma^2}\sum_{k=1}^{K} |\phi_k^\top \phi_k|^{\frac{1}{2}}$$

M-dis2= $$D_M(\mathcal{D};\mathbf{z},\boldsymbol{\phi}) = \sum_{i=1}^{n} |(\mathbf{x}_i - \mu_{z_i})^\top \Sigma^{-1}(\mathbf{x}_i - \mu_{z_i})|^{\frac{1}{2}}$$

Submission

- Implementation
  - Submit the code implementation before ?
  - Report 3 values in P10.
  - Submit as .tar file, including:

    1) Source Code with necessary comments, including

    1. Data generating
    2. DPM training, testing and evaluation

    2) ReadMe explaining

    1. How to generate data.
    2. Explain main functions, e.g., how to sample z and ϕ.
    3. How to run your code, from data preprocessing to training/testing/evaluation, step by step.
    4. what's the 3 values in P10
    5. If you do the Collapsed Gibbs sampling, write the derivation step by step in a file.
    6. If you do the Collapsed Gibbs sampling, compare the per-iteration time cost with Gibbs sampling and try to explain if it is always necessary to use the collapsed inference strategy and why.

Reference

[1]Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265.

[2]MacEachern, S. N. (1994). Estimating Normal Means With a Conjugate Style Dirichlet Process Procedure. *Communications in Statistics: Simulation and Computation*, 23, 727-741.

Thank You!