Unsupervised Learning of Distribution

Bor-shen Lin

bslin@cs.ntust.edu.tw

http://www.cs.ntust.edu.tw/~bslin

Learning of Distribution

- Unsupervised Learning (without output label)
- Given { $\mathbf{X_i}$ } → learn distribution $p(x)$
- Discrete variable
  - Learn probability weight function
  - $p(x)$ should satisfy $\sum_{-\infty}^{\infty} p(x) = 1$
- Continuous variable
  - Learn probability density function
  - $p(x)$ should satisfy $\int_{-\infty}^{\infty} p(x)dx=1$

Learning Models

- Parametric Model
  - Discrete distribution
  - Gaussian distribution
  - Gaussian mixture model
- Non-parametric Model (Instance-based Learning)
  - Nearest Neighbor Model
  - Kernel Model

Learning of Parametric Model

1. Learning of Discrete Distribution
2. Learning of Gaussian Distribution
3. Learning of Gaussian Mixture Model (GMM)

Estimation of Parameters

- θ represents a set of parameters for the probability density/mass function $P(X|\theta)$
- ***X*** is a set of i. i. d. observations X_1, X_2, …, X_n

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} P(\mathbf{X} \mid \boldsymbol{\theta})$$

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta} \mid \mathbf{X})$$

$$= \arg\max_{\boldsymbol{\theta}} \frac{P(\boldsymbol{\theta}, \mathbf{X})}{P(\mathbf{X})}$$

$$= \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta}) P(\mathbf{X} \mid \boldsymbol{\theta})$$

- Maximum Likelihood Estimation
- Maximum A Posteriori Estimation

1. Learning of Discrete Distribution

$\mathbf{X}$ *consists of* $X_1, X_2, \ldots, X_N$, *i.i.d. with p.w.f. as .*

$$P(X = v_k \mid \boldsymbol{\theta}) = w_k,\ k = 1, 2, \ldots, n$$

where $w_1 + w_2 + \ldots + w_n = 1$ *and* $\boldsymbol{\theta}$ *consists of* $w_1, w_2, \ldots, w_n$.

then $P(\mathbf{X} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} P(X_i \mid \boldsymbol{\theta})$,

$$= P(\mathbf{X} \mid w_1, w_2, \ldots, w_n) = w_1^{\,c_1} \cdot w_2^{\,c_2} \cdot \cdots \cdot w_n^{\,c_n},$$

where c_k *is the number of occurences for* v_k *in* $\mathbf{X}$

1. Estimation of Discrete Distribution

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} P(\mathbf{X} \mid \boldsymbol{\theta}) \text{ } \textit{is an optimization problem with constraint } w_1 + w_2 + \ldots + w_n = 1,$$

$$\textit{which can be solved with Lagrange multipler } L(X,\theta) \equiv P(\mathbf{X} \mid \boldsymbol{\theta}) + \lambda(\sum_{k=1}^{n} w_k - 1)$$

$$\nabla_\theta L(X,\theta) = 0, (\frac{\partial L}{\partial w_k} = 0 \ \forall w_k) \Rightarrow w_1^{c_1} \cdot w_2^{c_2} \cdots\cdots w_n^{c_n} \begin{pmatrix} c_1 / w_1 \\ \vdots \\ c_n / w_n \end{pmatrix} = -\lambda \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix},$$

$$\Rightarrow \frac{c_1}{w_1} = \frac{c_2}{w_2} = \ldots = \frac{c_n}{w_n} \equiv \eta \Rightarrow \sum_{k=1}^{N} c_k = \eta \sum_{k=1}^{n} w_k = \eta \quad (\textit{given} \sum_{k=1}^{n} w_k = 1)$$

$$\Rightarrow \hat{w}_k = \frac{c_k}{\eta} = \frac{c_k}{\sum_{k=1}^{N} c_k}$$

- Use occurrence count to estimate the probability weights for a p. w. f.

Constraint Optimization with Lagrange Multiplier

- Maximize $P(\boldsymbol{w})$ with the constraint: $\Sigma_m w_m = 1$

$$\hat{\mathbf{w}}_k = \frac{\mathbf{w}_k \dfrac{\partial P(\mathbf{w})}{\partial \mathbf{w}_k}}{\sum_{m=1}^{M} \mathbf{w}_m \dfrac{\partial P(\mathbf{w})}{\partial \mathbf{w}_m}}$$

2. Learning of Gaussian Distribution

$$X_1, X_2, \dots, X_N \ \textit{are i.i.d. with p.d.f. as}\ .P(X \mid \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \boldsymbol{\theta} = (\mu, \sigma^2)$$

$$\textit{then}\ P(\mathbf{X} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} P(X_i \mid \boldsymbol{\theta}),\ \mathbf{X}\ \textit{consists of}\ X_1, X_2, \dots, X_N$$

$$2\log(P(\mathbf{X} \mid \boldsymbol{\theta}) = 2\log(P(\mathbf{X} \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} 2\log(P(X_i \mid \boldsymbol{\theta}))$$

$$= \sum_{i=1}^{N} 2\log((2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}}) = -\sum_{i=1}^{N} \left(\log(2\pi\sigma^2) + \frac{(X_i-\mu)^2}{\sigma^2} \right)$$

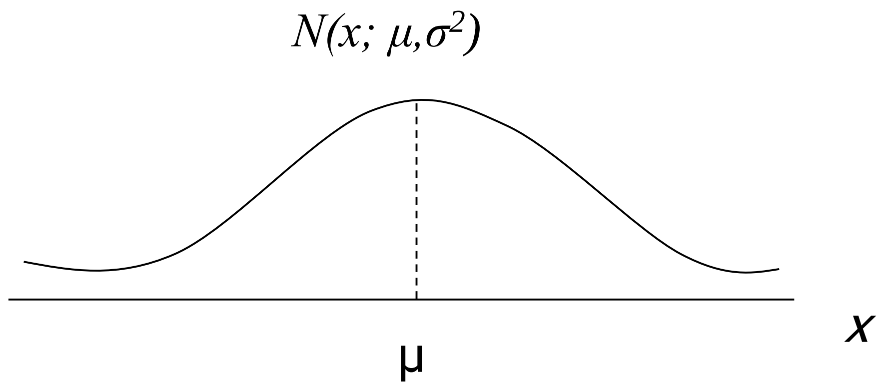
$$= -\sum_{i=1}^{N} \left(\log(2\pi) + \log(\sigma^2) + \frac{(X_i-\mu)^2}{\sigma^2} \right) = -N\log(2\pi) - \sum_{i=1}^{N} \left(\log(\sigma^2) + \frac{(X_i-\mu)^2}{\sigma^2} \right)$$

$$l(\boldsymbol{\theta}) \equiv 2\log(P(\mathbf{X} \mid \boldsymbol{\theta}) + N\log(2\pi)\ \textit{is monotonic with}\ P(\mathbf{X} \mid \boldsymbol{\theta})$$

Gaussian Distribution

$$N(x;\mu,\sigma^2) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{(x-\mu)^2}{\sigma^2}\right)}$$

$$E(X) = \mu, Var(X) = \sigma^2$$



Estimation of Gaussian Distribution

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} P(\mathbf{X} \mid \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$$

$$\frac{\partial l(\mu, \sigma^2)}{\partial \mu} = 0, \frac{\partial}{\partial \mu}\left[\sum_{i=1}^{N} \frac{(X_i - \mu)^2}{\sigma^2}\right] = 2\sum_{i=1}^{N}(X_i - \mu) = 0$$

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} X_i$$

- Every observation X_i contributes to the estimation of μ(weight as 1/N)

$$\frac{\partial l(\mu, \sigma^2)}{\partial \sigma^2} = 0, \frac{\partial}{\partial \sigma^2}\sum_{i=1}^{N}\left[\log(\sigma^2) + \frac{(X_i - \mu)^2}{\sigma^2}\right] = 0$$

$$\sum_{i=1}^{N}\left[\frac{1}{\sigma^2} - \frac{(X_i - \mu)^2}{\sigma^4}\right] = 0, N\sigma^2 = \sum_{i=1}^{N}\left(X_i - \mu\right)^2$$

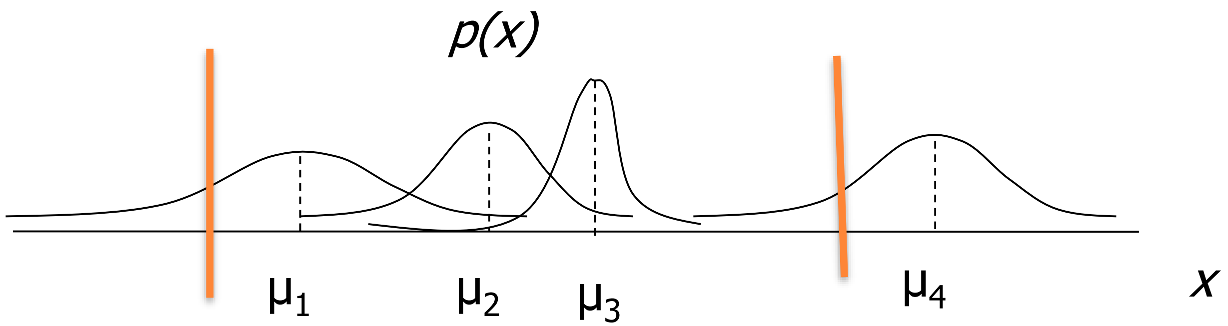
$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}\left(X_i - \mu\right)^2$$

- Every observation X_i contributes to the estimation of σ^2(weight as 1/N)

Gaussian Mixture Model (GMM)

$$p(x) = \sum_{k=1}^{M} c_k N(x; \mu_k, \sigma_k{}^2)$$

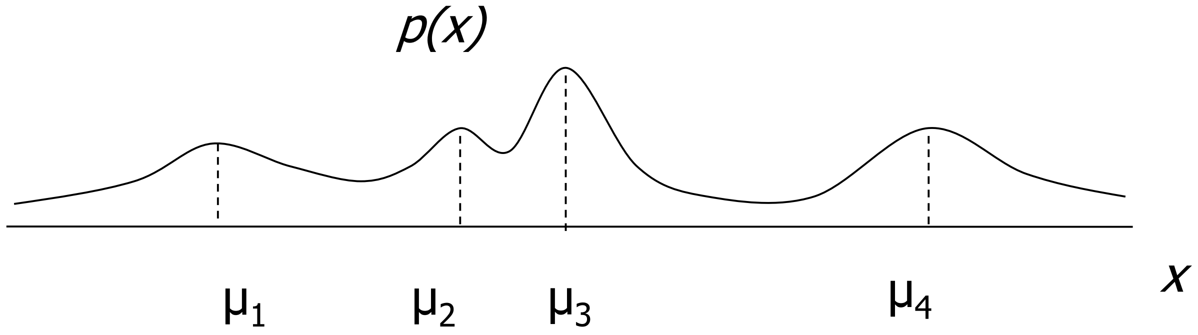
$$\int_{-\infty}^{\infty} p(x)dx = \sum_{k=1}^{M} c_k \int_{-\infty}^{\infty} N(x; \mu_k, \sigma_k{}^2)dx = \sum_{k=1}^{M} c_k = 1.0$$



Partition Gaussian Model(PGM)

$$p(x) \equiv \max_k \; N(x; \mu_k, {\sigma_k}^2)$$

$$\int_{-\infty}^{\infty} p(x)dx \neq 1.0$$



Multi-Dimensional Gaussian Distribution

$$N(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathbf{t}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

$$\boldsymbol{\mu} \equiv E(\mathbf{X}) = \int \mathbf{x} \cdot P(\mathbf{x}) \cdot d\mathbf{x} \quad n \times 1$$

$$\boldsymbol{\Sigma} \equiv E((\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})^t) \quad n \times n$$

- $\boldsymbol{x}$: n dimensional vector
- Each Gaussian: $\boldsymbol{\mu}$ as *mean vector*, $\boldsymbol{\Sigma}$ as *covariance matrix*
- Stochastically independent when $\boldsymbol{\Sigma}$ is diagonal
- Multi-dimensional GMM: $\theta = \{(c_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\}$
  - # of parameters : $M(1+n+n^2)$

3. Learning of GMM with Expectation Maximization (EM)

$$X_1, X_2, \ldots, X_N \text{ are i.i.d. with p.d.f. } P(X \mid \boldsymbol{\theta}) = \sum_{m=1}^{M} c_m \cdot P(X; \mu_m, \sigma_m{}^2),$$

$$\text{where } n(X; \mu_m, \sigma_m{}^2) = \frac{e^{-\frac{(x-\mu_m)^2}{2\sigma_m{}^2}}}{\sqrt{2\pi}\sigma_m}, \boldsymbol{\theta} = \{c_m, \mu_m, \sigma_m{}^2\} \text{ and } \sum_{m=1}^{M} c_m = 1.$$

$$\text{Then } P(\mathbf{X} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} P(X_i \mid \boldsymbol{\theta}), \mathbf{X} \text{ consists of } X_1, X_2, \ldots, X_N.$$

$$\text{The Lagrange function is } L(\boldsymbol{\theta}) = \log P(\mathbf{X} \mid \boldsymbol{\theta}) + \lambda\left(\sum_{m=1}^{M} c_m - 1\right)$$

$$= \sum_{i=1}^{N} \log P(X_i \mid \boldsymbol{\theta}) + \lambda\left(\sum_{m=1}^{M} c_m - 1\right).$$

Re-estimation of GMM Parameters

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \mu_m} = \sum_{i=1}^{N} \frac{c_m \cdot \dfrac{\partial P(X_i; \mu_m, \sigma_m{}^2)}{\partial \mu_m}}{P(X_i \mid \boldsymbol{\theta})} = \sum_{i=1}^{N} \frac{c_m \cdot P(X_i; \mu_m, \sigma_m{}^2) \cdot \dfrac{(X_i - \mu_m)}{\sigma_m{}^2}}{P(X_i \mid \boldsymbol{\theta})} = 0$$

$$P(X_i, C_m) \equiv c_m \cdot P(X_i; \mu_m, \sigma_m{}^2), \boldsymbol{\theta}_m \equiv (c_m, \mu_m, \sigma_m),$$

$$l(m,i) \equiv \frac{P(X_i, C_m)}{P(X_i)} = P(C_m \mid X_i)$$

$$\Rightarrow \sum_{i=1}^{N} l(m,i)(X_i - \mu_m) = 0 \Rightarrow \mu_m \sum_{i=1}^{N} l(m,i) = \sum_{i=1}^{N} l(m,i) X_i$$

$$\Rightarrow \hat{\mu}_m = \frac{\sum_{i=1}^{N} l(m,i) X_i}{\sum_{i=1}^{N} l(m,i)}$$

- $l(m,i)$: estimated probability that X_i is produced by m-th mixture (weight)
- Denominator for normalization

Re-estimation of GMM Parameters

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \sigma^2{}_m} = \sum_{i=1}^{N} \frac{c_m \cdot \dfrac{\partial P(X_i; \mu_m, \sigma_m{}^2)}{\partial \sigma^2{}_m}}{P(X_i \mid \boldsymbol{\theta})} = 0$$

$$\sum_{i=1}^{N} \frac{P(X_i \mid \boldsymbol{\theta}_m) \cdot (1 - \dfrac{(X_i - \mu_m)^2}{\sigma_m{}^2})}{P(X_i \mid \boldsymbol{\theta})} = 0$$

$$\Rightarrow \sum_{i=1}^{N} l(m,i) = \sum_{i=1}^{N} l(m,i) \frac{(X_i - \mu_m)^2}{\sigma_m{}^2}$$

$$\Rightarrow \hat{\sigma}_m{}^2 = \frac{\sum_{i=1}^{N} l(m,i)(X_i - \mu_m)^2}{\sum_{i=1}^{N} l(m,i)}$$

CONCEPT

- μ_m is a independent variable that may be adjusted freely no matter what others parameters are.
- μ_m has a unique global maximum.
- The global maximum is located at $\frac{\partial L(\theta)}{\partial \mu_m} = 0$.
- A better value of μ_m is guaranteed independent through the iteration formula of $\hat{\mu}_m$.
- Both μ_m and σ_m are independent variables.
- c_m's are dependent variables, since $\sum_m c_m = 1$.

Re-estimation of GMM Parameters

$$\frac{\partial L(\boldsymbol{\theta})}{\partial c_m} = \sum_{i=1}^{N} \frac{P(X_i; \mu_m, \sigma_m{}^2)}{P(X_i \mid \boldsymbol{\theta})} + \lambda = 0 \quad \forall m$$

$$P(X_i \mid \boldsymbol{\theta}_m) \equiv c_m \cdot P(X_i; \mu_m, \sigma_m{}^2), \boldsymbol{\theta}_m \equiv (c_m, \mu_m, \sigma_m), l(m,i) \equiv \frac{P(X_i \mid \boldsymbol{\theta}_m)}{P(X_i \mid \boldsymbol{\theta})}$$

$$\Rightarrow \sum_{i=1}^{N} \frac{P(X_i; \mu_m, \sigma_m{}^2)}{P(X_i \mid \boldsymbol{\theta})} = \sum_{i=1}^{N} \frac{P(X_i \mid \boldsymbol{\theta}_m)}{c_m P(X_i \mid \boldsymbol{\theta})} = \frac{\sum_{i=1}^{N} l(m,i)}{c_m} = \varepsilon \ \forall m$$

$$\Rightarrow 1 = \sum_{m=1}^{M} c_m = \sum_{m=1}^{M} \frac{\sum_{i=1}^{N} l(m,i)}{\varepsilon} \Rightarrow \varepsilon = \sum_{m=1}^{M} \sum_{i=1}^{N} l(m,i)$$

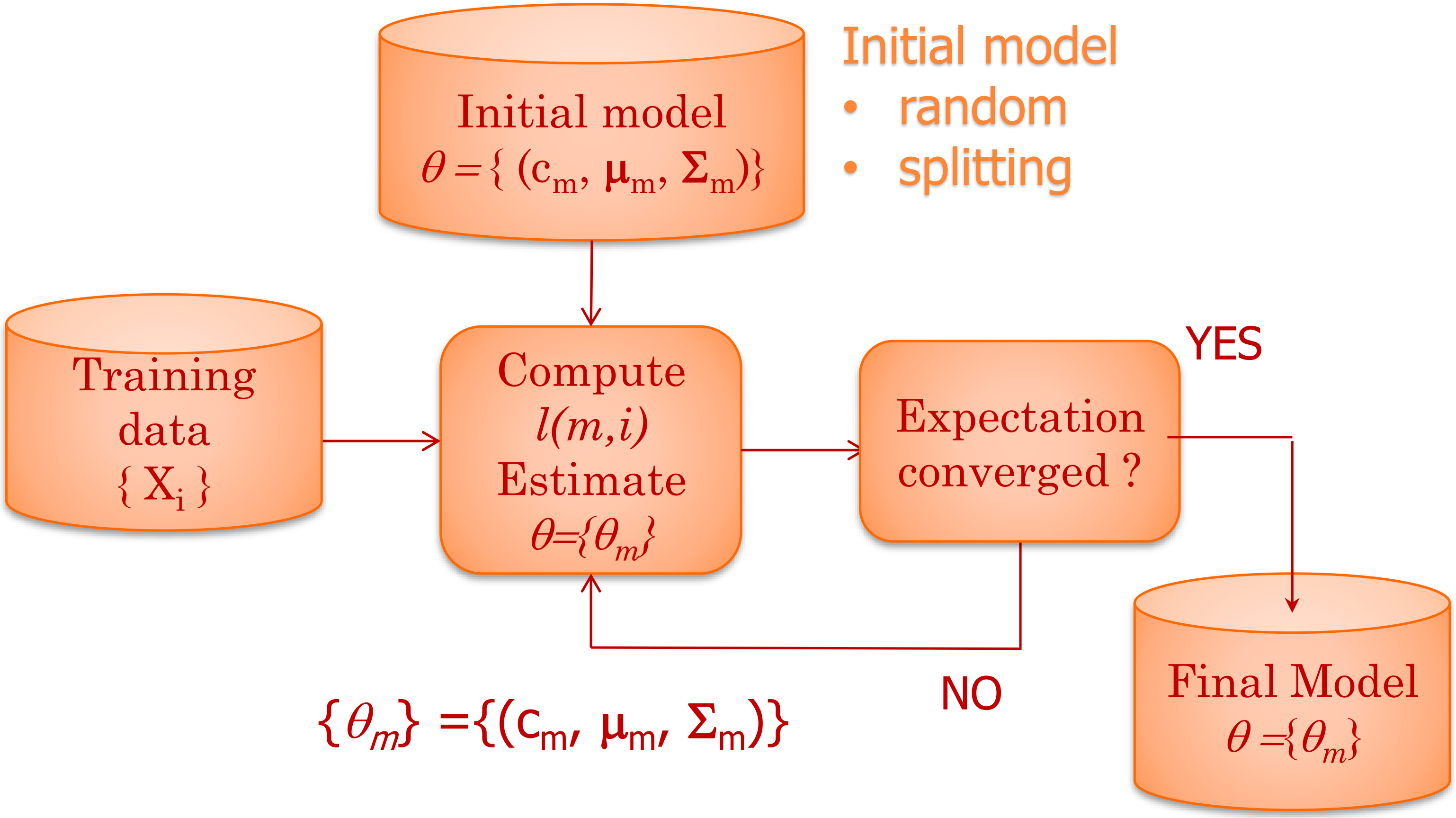
$$\Rightarrow \hat{c}_m = \frac{\sum_{i=1}^{N} l(m,i)}{\varepsilon} = \frac{\sum_{i=1}^{N} l(m,i)}{\sum_{m=1}^{M} \sum_{i=1}^{N} l(m,i)}$$

EM Reestimates of GMM Parameters (Multi-Dimensional)

- $\hat{\boldsymbol{\mu}}_m = \frac{\sum_i l(m,i)\boldsymbol{x}_i}{\sum_i l(m,i)}$
- $\hat{\boldsymbol{\Sigma}}_m = \frac{\sum_i l(m,i)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^t}{\sum_i l(m,i)}$
- $c_m = \frac{\sum_i l(m,i)}{\sum_m \sum_i l(m,i)}$

Learning for GMM

Initial model
- random
- splitting

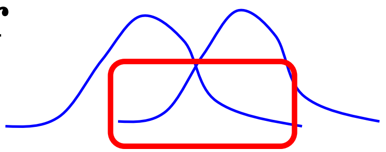


Steps of GMM Training

1. Set the initial mixture as the mean and covariance of all training data $\{X_i\}$ $(M = 1)$.
2. Split the largest cluster into two clusters
   - from the mean, equal weights
   - Other algorithm: LBG(1→2→4→8→…)
3. Re-estimate the model parameters iteratively until converged.
4. Repeat steps 2 & 3 till M mixtures.

Distance Between Gaussians - 1

- Bhattacharyya Divergence $D_B(f, g)$
  - $D_B(f, g)$ estimation of classification error



$$D_B(f,g) \equiv -\log \int \sqrt{f(x)g(x)}\,dx$$

$$= \frac{1}{4}(\boldsymbol{\mu}_f-\boldsymbol{\mu}_g)^t(\boldsymbol{\Sigma}_f+\boldsymbol{\Sigma}_g)^{-1}(\boldsymbol{\mu}_f-\boldsymbol{\mu}_g)+\frac{1}{2}\log\left|\frac{\boldsymbol{\Sigma}_f+\boldsymbol{\Sigma}_g}{2}\right| - \frac{1}{4}\log\left|\boldsymbol{\Sigma}_f\boldsymbol{\Sigma}_g\right|$$

$$\textit{Then Bayes error } B_e(f,g) \equiv \frac{1}{2}\int \min(f(x),g(x))\,dx \le \boxed{\frac{1}{2}e^{-D_B(f,g)}}$$

Distance Between Gaussians - 2

- Kullback-Leibler Divergence (KLD)

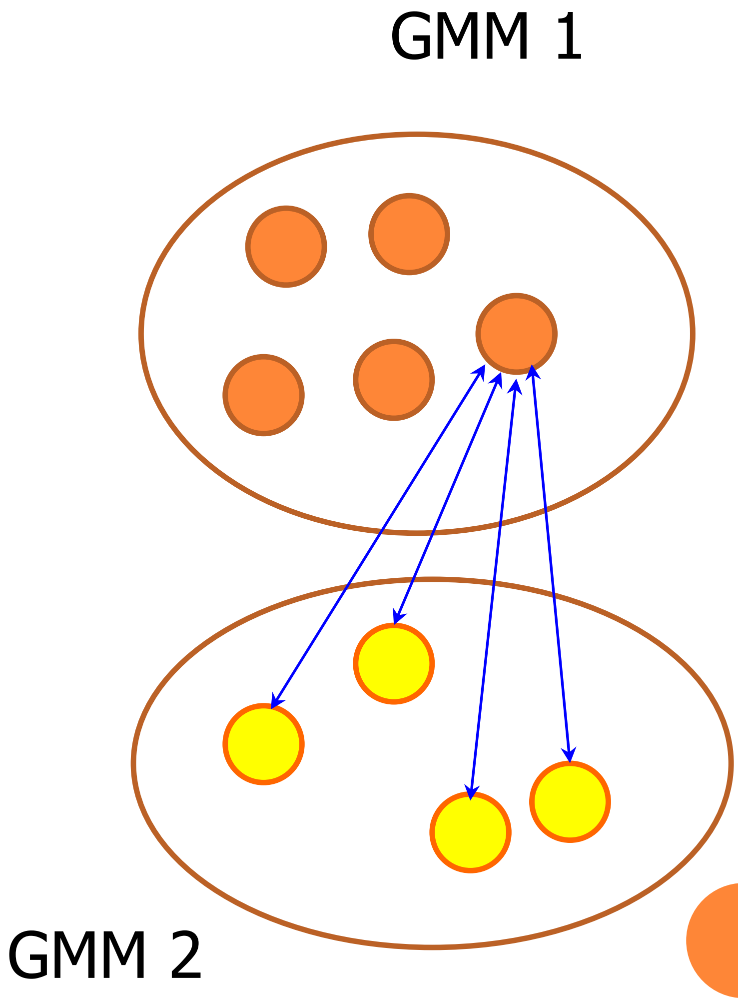
$$D_{KL}(f,g) \equiv \int f(x)\log\frac{f(x)}{g(x)}dx$$

$$= \frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_g|}{|\boldsymbol{\Sigma}_f|} + Tr\left|\boldsymbol{\Sigma}_g{}^{-1}\boldsymbol{\Sigma}_f\right| - d + (\boldsymbol{\mu}_f - \boldsymbol{\mu}_g)^t(\boldsymbol{\Sigma}_f - \boldsymbol{\Sigma}_g)^{-1}(\boldsymbol{\mu}_f - \boldsymbol{\mu}_g)\right]$$

$$Then\ D_{KL}(f,g) \geq 2D_B(f,g)$$

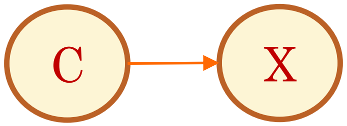
Similarity between Two GMMs

- Pairwise distances
  - Bhattacharyya distance
- Weighted average



Applications of GMM

- Parametric model of continuous variables
  - Unsupervised learning of $p(X)$
- Clustering (unsupervised)
  - Regarding every mixture as a cluster
- Classification (supervised)
  - Train $p(X|C_m)$ for all m's (classes)
- Examples
  - Language identification
  - Gender identification
  - Speaker recognition
  - Image classification/tagging



GMM-Based Clustering

1. Every mixture of a GMM regarded as a cluster
   - Similar to k-Means clustering, however the variances are used for distance normalization when computing the probability (exponential term)

     (simple k-Means uses Euclidean distance)
   - A GMM is trained, and each point is assigned to a cluster according to:

$$k^* = argmax_k(l(X,k) = argmax_k\left(\frac{c_k p_k(X)}{\sum_i c_i p_i(X)}\right)$$

2. A GMM is regarded as a point
   - Clustering of GMMs based on distances
   - Example: speaker clustering (groups)

     training GMMs for all speakers

GMM-based Classifier

- Train GMMs of $p(\boldsymbol{x} \mid C_i)$ for i=0,1 respectively
- ML Detector

$$C^* = argmax_i \, p(\boldsymbol{x} \mid C_i)$$

- MAP Detector (Given the prior distribution)

$$C^* = argmax_i \, p(C_i \mid \boldsymbol{x})$$
$$= argmax_i \, p(C_i) p(\boldsymbol{x} \mid C_i)$$

Discriminative Training for GMM

- ML training
  - The objective functions to be maximized is the likelihood function for every class
  - Every GMM are trained with the data of its class
  - A sample of class k will influence the distribution of that class, i.e. $p(\boldsymbol{x}\,|\,C_k)$, only
- Minimum classification error (MCE) training
  - The objective function to be minimized is the overall classification errors
  - The GMMs for different classes are trained jointly instead of individually
  - Every sample will influence the distributions of all classes, i.e. $p(\boldsymbol{x}\,|\,C_j)$ *for all* j.

MCE Training

- $p_k(x)$ is a GMM with parameters $\{ (c_{km}, \mu_{km}, \Sigma_{km})\}$

  $p_k(x) = \sum_{m=1}^{M} c_{km} p_{km}(x)$
- $g_k(x) = \log(p_k(x))$
- $d_k(x) = -g_k(x) + g_{\bar{k}}(x)$

  $$= -g_k(x) + log\left[\frac{1}{M-1}\left\{\sum_{j \neq k} e^{\eta g_j(x)}\right\}\right]^{1/\eta}$$
- $l_k(x) = \frac{1}{1+e^{-\gamma d_k + \theta}}$ (sigmoid)
- $L(X) = \sum_{k=1}^{K} \sum_{x_i \in C_k} l_k(x_i)$
  - Minimizing l leads to the minimization of classification errors
  - The parameters can be obtained by gradient probabilistic descent (GPD) $d\Lambda = -\epsilon \nabla L$

MCE Formula – Diagonal Covariance

- For $x_i \in C_k, \theta_{jm} \equiv \frac{c_{jm} p_{jm}}{p_j}, r_k \equiv \gamma l_k (1 - l_k)$
- $d\mu_{kml} = \varepsilon r_k \theta_{km} \frac{x_l - \mu_{kml}}{{\sigma_{kml}}^2}$

  $d\mu_{jml} = -\varepsilon r_k \frac{p_j}{\sum_{n \neq k} p_n} \theta_{jm} \frac{x_l - \mu_{jml}}{{\sigma_{jml}}^2} \, for \; j \neq k$
- $d\sigma_{kml} = \varepsilon r_k \theta_{km} \frac{1}{\sigma_{kml}} \left(\frac{(x_l - \mu_{kml})^2}{{\sigma_{kml}}^2} - 1 \right)$ $d\sigma_{jml} =$

  $- \varepsilon r_k \frac{p_j}{\sum_{n \neq k} p_n} \theta_{jm} \frac{1}{\sigma_{jml}} \left(\frac{(x_l - \mu_{jml})^2}{{\sigma_{jml}}^2} - 1 \right) for \; j \neq k$
- *Minimum classification error rate for speech recognition*, IEEE Trans. on Speech and Audio Processing, 1997.

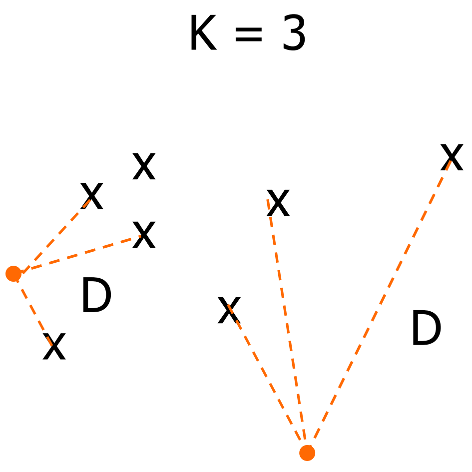
INSTANCE-BASED LEARNING

- Weakness of parametric model
  - restricted family of function might over-simplify the real world
- Non-parametric learning
  - All samples are stored and used for model
  - Instance-based learning or memory-based learning
  - Complexity is increased as the data set grows.
- Estimation of p(x)
  - A type of unsupervised learning

1. Nearest Neighbor Model
2. Kernel Model

Nearest-neighbor Models

- Estimation of density
  - Use the largest distance for the k nearest-neighbors
  - The larger the distance, the lower the density of the point $\boldsymbol{x}$
  - k is low → $p(\boldsymbol{x})$ highly variable
    k is large → $p(\boldsymbol{x})$ smooth
- Distance measure
  - Euclidean distance might not be appropriate (e.g. D = $\boldsymbol{d}^t\Sigma^{-1}\boldsymbol{d}$)
  - Should consider the physical meanings of different dimensions



$$p(\boldsymbol{x}) \propto \frac{k}{D^n}$$

Kernel Models

- $p(x)$ is estimated with the normalized sum of the kernel functions for all training instances $\{\boldsymbol{x}_i\}$
- $p(x) = \frac{1}{N}\sum_i K(\boldsymbol{x}, \boldsymbol{x}_i)$
  - $K(\boldsymbol{x}, \boldsymbol{x}_i)$ is the measure of similarity that depends on $D(\boldsymbol{x}, \boldsymbol{x}_i)$
  - A popular kernel: $K(\boldsymbol{x}, \boldsymbol{x}_i) = \frac{1}{\sqrt{2\pi w^2}^d} e^{-\frac{D(\boldsymbol{x},\boldsymbol{x}_i)^2}{2w^2}}$