Decision theory. Fisher's LDA, QDA. Mixtures. EM algorithm

V. Kitov

Yandex School of Data Analysis

Imperial College London
Department of Physics

January 2015

Table of Contents

Bayes Decision Rule for Minimum Error

- C classes: $\omega_1, \omega_2, ..., \omega_C$.
- Need to predict class for target object.

Initial solution (features unknown)

$$\widehat{c} = \arg\max_c p(\omega_c)$$

- Now features x of object are observed.

Solution after observing x

$$\widehat{c} = \arg\max_c p(\omega_c|x)$$

Reformulation

Rewrite class posterior probability:

$$p(\omega_c|x) = \frac{p(\omega_c, x)}{p(x)} = \frac{p(\omega_c)p(x|\omega_c)}{p(x)}$$

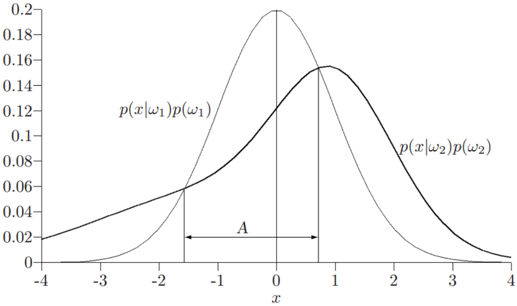
Reformulated solution after observing x

$$\widehat{c} = \arg\max_c p(\omega_c)p(x|\omega_c)$$

Two class case:

$$\text{Assign to class } \omega_1 \text{ if } \frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{p(\omega_2)}{p(\omega_1)}$$

Illustration of decision rule



Class ω_1 is selected in region A, class ω_2 selected outside of A

Minimum error property

Let $\Omega_1, \Omega_2, ...\Omega_C$ be decision regions for classes $\omega_1, \omega_2, ...\omega_C$ and Ω denotes the whole features space.
Then probability of correct classification equals:

$$p(correct) = \sum_{c=1}^{C} p(correct, \omega_c) = \sum_{c=1}^{C} p(x \in \Omega_c, \omega_c) =$$

$$= \sum_{c=1}^{C} \int_{x \in \Omega_c} p(x, \omega_c) dx = \sum_{c=1}^{C} \int_{x \in \Omega_c} p(x) p(\omega_c | x) dx$$

Probability of correct classification is maximized when for each c $\Omega_c = \{x : c = \arg\max_i p(\omega_i | x)\}$ - Bayes rule!

$$p(error) = 1 - p(correct)$$

Probability of error is minimized for Bayes decision rule.

Probabilities for Bayes minimum error decision rule

By definition of Bayes decision rule for each c:

$$p(correct, \omega_c) = \int_{\Omega_c} p(\omega_c)p(x|\omega_c)dx = \int_{\Omega_c} \max_i p(\omega_i)p(x|\omega_i)dx$$

Probability of correct classification equals (using property $\bigcap_c \Omega_c = \emptyset$, $\bigcup_c \Omega_c = \Omega$):

$$p(correct) = \sum_{c=1}^{C} p(correct, \omega_c) = \int \max_i p(\omega_i)p(x|\omega_i)dx$$

Probability of erroneous classification:

$$p(error) = 1 - \int \max_i p(\omega_i)p(x|\omega_i)dx$$

Reject option

- Most of the misclassification errors occurs when algorithm is unsure:

$$p(\omega_{\widehat{c}}|x) \text{ is small, where } \widehat{c} = \arg\max_c p(\omega_c)$$

- Introduce reject option: if $\max_c p(\omega_c)$ is small, then reject classification and
  - leave as third class
  - classify is later
    - when new information about object becomes available
    - classify using different algorithm

Rejection region

Rejection region

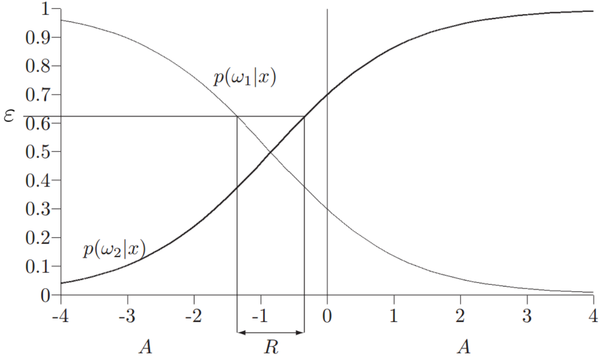
$$R = \{x : \max_c p(\omega_c|x) < \varepsilon\}$$

Acceptance region

$$A = \{x : \max_c p(\omega_c|x) \geq \varepsilon\}$$

Trade-off: larger reject region will decrease error-rate, but more otherwise correctly classified objects will be rejected.

Illustration of rejection region



Acceptance and rejection regions.

Probabilities

Correct classification with reject option implies:

- object was not rejected
- object was correctly classified

Probability of correct classification:

$$p(correct) = \int_A \max_i p(\omega_i)p(x|\omega_i)dx$$

Probability of rejection:

$$p(reject) = \int_R p(x)dx$$

Probability of error:

$$p(error) = \int_A (1 - \max_i p(\omega_i)p(x|\omega_i))dx = 1 - p(reject) - p(correct)$$

Table of Contents

Motivation

- In most practical situations loss from misclassifying different classes is different.
- Examples:
  - e-mail filtering: spam/not spam
  - medical diagnostics: ill/healthy
  - bank crediting: reliable borrower/unreliable borrower
  - network intrusion detection: intrusion/usual use
- Possible solution:

Assign different costs to different misclassifications.

Derivation

- Define $C\text{x}C$ loss matrix $\mathbf{\Lambda}$, where $\lambda_{i,j}$ equals cost when object belongs to class ω_i but was assigned to class ω_j.
- Average loss equals

$$\mathbf{E}[L] = \sum_i \sum_j \lambda_{ij} p(\omega_i, \widehat{\omega}_j) =$$
$$= \sum_i \sum_j \lambda_{ij} \int_{\Omega_j} p(x, \omega_i) dx = \sum_j \int_{\Omega_j} \sum_i \lambda_{ij} p(x, \omega_i) dx$$

- It follows that minimum average solution satisfies:

$$\Omega_c = \{x : \; c = \arg\min_j \sum_i \lambda_{ij} p(x, \omega_i)$$
$$= \arg\min_j \sum_i \lambda_{ij} p(\omega_i | x) p(x) = \arg\min_j \sum_i \lambda_{ij} p(\omega_i | x)\}$$

Bayes minimum risk classification

Bayes minimum risk classification

$$\widehat{c} = \arg\min_{c} \sum_{i} \lambda_{ic} p(\omega_i | x)$$

Average risk attained by this decision rule equals:

$$\mathbf{E}[L] = \sum_{i} \sum_{j} \lambda_{ij} p(\omega_i, \widehat{\omega}_j) =$$

$$= \sum_{i} \sum_{j} \lambda_{ij} \int_{\Omega_j} p(x, \omega_i) dx = \sum_{j} \int_{\Omega_j} \sum_{i} \lambda_{ij} p(x, \omega_i) dx$$

$$= \int \min_{j} \sum_{i} \lambda_{ij} p(x, \omega_i) dx$$

Bayes minimum risk classification

Two class case for $\lambda_{11} = \lambda_{22} = 0$:

$$\text{Assign to class } \omega_1 \text{ if } \frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{\lambda_{21} p(\omega_2)}{\lambda_{12} p(\omega_1)}$$

For zero-one loss matrix

$$\lambda_{ij} = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases}$$

Bayes minimum risk solution reduces to Bayes minimum error solution.

Reject option

Define conditional risk of assigning object x to class ω_j:

$$loss^j(x) = \sum_i \lambda_{ij} p(\omega_i|x)$$

For Bayes minimum risk decision rule expected loss for object x is

$$loss(x) = \min_j loss^j(x)$$

We can reject classification if expected cost of misclassification object is too high.

$$\begin{aligned} \text{rejection region:} \quad & R = \{x : \; loss(x) > \varepsilon\} \\ \text{acceptance region:} \quad & A = \{x : \; loss(x) \leq \varepsilon\} \end{aligned}$$

Reject option

Expected risk with rejection option equals:

$$\mathbf{E}[L|\text{no rejection}] = \int_A \min_j \sum_i \lambda_{ij} p(x, \omega_i) dx$$

Probability of rejection:

$$p(\text{reject}) = \int_R p(x) dx$$

Bayes minimum risk for equal within class costs

Earlier we have obtained Bayes minimum risk decision rule:

$$\widehat{c} = \arg\min_c \sum_i \lambda_{ic} p(\omega_i|x)$$

Suppose the costs matrix Λ has simpler structure:

$$\lambda_{ic} = \begin{cases} 0, & i = c \\ \gamma_i, & i \neq c \end{cases}$$

Then Bayes minimum risk becomes:

$$\begin{aligned} \widehat{c} &= \arg\min_c \sum_{i \neq c} \gamma_i p(\omega_i|x) = \arg\min_c \left\{ \sum_i \gamma_i p(\omega_i|x) - \gamma_c p(\omega_c|x) \right\} \\ &= \arg\max_c \gamma_c p(\omega_c|x) \end{aligned}$$

Table of Contents

Neyman-Pearson rule

- Applied for two class cases. Class ω_1 will be the target class
- Two types of errors incurred:
  - False alarm: $p(\widehat{\omega}_1|\omega_2)$
  - Missed detection: $p(\widehat{\omega}_2|\omega_1)$
- Instead of maximizing general performance we minimize false alarm rate, given tolerable missed detection rate.

$$\begin{cases} \int_{\Omega_2} p(x|\omega_1) \to \min_{\Omega_1} \\ \int_{\Omega_1} p(x|\omega_2) = \varepsilon \end{cases}$$

- Neyman-Pearson solution:

Assign to class ω_1 if $\dfrac{p(x|\omega_1)}{p(x|\omega_2)} > \mu$, where μ is such that $\displaystyle\int_{\Omega_1} p(x|\omega_2) = \varepsilon$

Minimax criterion

- Bayes minimum error rule:

$$p(error) = p(\omega_1) \int_{\Omega_2} p(x|\omega_1)dx + p(\omega_2) \int_{\Omega_1} p(x|\omega_2)dx$$
$$= p(\omega_1) \int_{\Omega_2} p(x|\omega_1)dx + (1 - p(\omega_1)) \int_{\Omega_1} p(x|\omega_2)dx$$

- Suppose, prior class probabilities may change in future.
- Optimize Bayes rule for worst case of prior probabilities:

$$\max_{p(\omega_1)} p(error) \to \min_{\Omega_1}$$

Minimax criterion

- $p(error)$ is a linear function of $p(\omega_1)$, so maximum is achieved on the edge and is equal to
$$\max\left\{\int_{\Omega_2} p(x|\omega_1)dx, \int_{\Omega_1} p(x|\omega_2)dx\right\}$$
- Solution:
$$\int_{\Omega_2} p(x|\omega_1)dx = \int_{\Omega_1} p(x|\omega_2)dx$$

Table of Contents

Definition

Discriminant functions approach

- have C discriminant functions $g_i(x)$, $i = 1, 2 \ldots C$.
- assign x to class having maximum discriminant function value:

$$\widehat{c} = \arg\max_c g_c(x)$$

Discriminant functions are not unique:
$g_i(x)$ and $g'(x) = f(g(x))$ lead to equivalent classification for any monotonically increasing function $f(x)$.

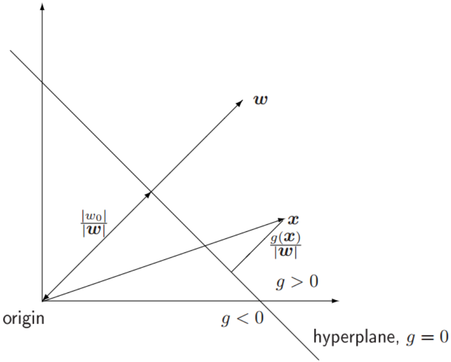
Two class case

- For two class case we may define a single function $g(x)$ such that
$$\widehat{c} = \begin{cases} 1, & g(x) > k, \\ 2 & g(x) \leq k. \end{cases}$$
- Possible reductions from multiclass case:
  - $g(x) = g_1(x) - g_2(x)$, $k = 0$
  - $g(x) = g_1(x)/g_2(x)$, $k = 1$ for positive $g_1(x), g_2(x)$.

Linear discriminant function

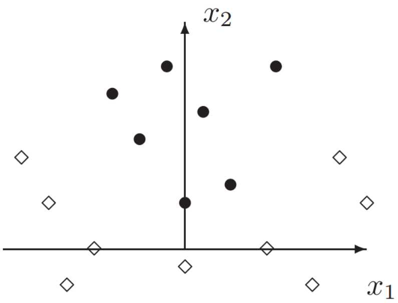
Simplest case - linear discriminant function:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$$



Limitation of linear decision boundary

The objects below can't be separated with linear boundary.



However, objects may be linearly separated in transformed space: $\phi_1(\boldsymbol{x}) = x_1^2$, $\phi_2(\boldsymbol{x}) = x_2$.

Non-linear discriminant functions

Natural way to make non-linear decision boundaries is to apply standard linear discriminant functions with transformed features.

Most well-known examples:

- linear: $\phi_i(\boldsymbol{x}) = x_i$
- polynomial: $\phi_i(\boldsymbol{x}) = x_{k_1}^{s_1} x_{k_2}^{s_2} ... x_{k_q}^{s_q}$
- radial basis functions: $\phi_i(\boldsymbol{x}) = \phi(|\boldsymbol{x} - \boldsymbol{\nu_i}|)$, where $\phi(\cdot)$ is non-increasing function, meaning proximity.
- multi-layer perceptron: $\phi_i(\boldsymbol{x}) = f(\boldsymbol{x}^T \boldsymbol{\nu}_i + \nu_{i0})$, where $f(z) = 1/(1+e^{-z})$ - logistic or any other “step” function.

Probability calibration

- Assume we have an arbitrary discrimination function $g(x)$ and two classes.
- Classification is based on the sign: $y = \operatorname{sign} g(x)$
- Need to estimate posterior class probabilities

$$p(\omega_1|x) = F(R_\theta(g(x)))$$

where $R_\theta(z)$ is monotone transform and F maps $\mathbb{R}$ to $[0, 1]$.
- Can assume $F(z) = \sigma(z)$, $R_\theta(z) = \theta_0 + \theta_1 z$ (Platt's calibration)
- Then, using the property $1 - \sigma(z) = \sigma(-z)$:

$$p(y = 1|x) = \sigma\left(\theta_0 + \theta_1 g(x)\right), \quad p(y = -1|x) = \sigma\left(-\theta_0 - \theta_1 g(x)\right)$$

- Estimate using ML:

$$\prod_{i=1}^{n} \sigma\left(y_i(\theta_0 + \theta_1 g(x))\right) \to \max_\theta$$

Table of Contents

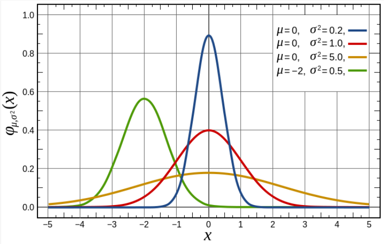
Gaussian (normal) distribution

Univariate case:

$$p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Multivariate case (D-dimensionality of data):

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$



Gaussian (normal) distribution - sample estimates

Univariate case:

$$\hat{\mu}_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \hat{\sigma}^2_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \widehat{\mu})^2$$

Multivariate case (D-dimensionality of data):

$$\hat{\mu}_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \widehat{\Sigma}_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \widehat{\mu})(x_n - \widehat{\mu})^T$$

Since $\widehat{\sigma}^2_{ML}$ and $\widehat{\Sigma}_{ML}$ are biased estimates, it is common to use unbiased estimates:

$$\hat{\sigma}^2_{ML} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \widehat{\mu})^2, \quad \widehat{\Sigma}_{ML} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \widehat{\mu})(x_n - \widehat{\mu})^T$$

Gaussian classifier

- In Gaussian classifier

$$p(x|\omega_j) = \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma_j^{-1}(x-\mu)\right\}$$

- It follows that

$$\begin{aligned}\log p(x|\omega_j) &= \log p(x|\omega_j) + \log p(\omega_j) - \log p(x) \\ &= -\frac{1}{2}(x-\mu_j^T)\Sigma_j^{-1}(x-\mu_j) - \frac{1}{2}\log|\Sigma_j| \\ &\quad -\frac{d}{2}\log(2\pi) + \log p(\omega_j) - \log p(x)\end{aligned}$$

- Removing common additive terms, we obtain discriminant functions:

$$g_j(x) = \log p(\omega_j) - \frac{1}{2}\log|\Sigma_j| - \frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j) \quad (1)$$

Practical application

- In practice we replace theoretical terms μ_j, Σ_j with their sample estimates $\widehat{\mu}_j$, $\widehat{\Sigma}_j$.
- $p(\omega_j)$ may be estimated using sample frequency (assuming that experiment conditions don't change): $p(\omega_j) = \frac{n_j}{n}$.

$$g_j(x) = \log p(\omega_j) - \frac{1}{2}\log|\widehat{\Sigma}_j| - \frac{1}{2}(x - \widehat{\mu}_j)^T \widehat{\Sigma}_j^{-1}(x - \widehat{\mu}_j)$$

- Analysis:
  - depends on normality assumptions (in particular - on unimodality)
  - needs to specify:
    - CD parameters to estimate $\widehat{\mu}_j$, $j = 1, 2, ...C$.
    - $CD(D+1)/2$ parameters to estimate $\widehat{\Sigma}_j$, $j = 1, 2, ...C$.

Simplifying assumptions

- $CD(D+3)/2$ may be too large for multidimensional tasks with small training sets.
- Simplifying assumptions:
  - **Naive Bayes**: assume that $\Sigma_1, \Sigma_2, ...\Sigma_C$ are diagonal.
  - **Project onto a subspace**: for example on first few principal components.
  - **Proportional covariance matrices**: assume that $\Sigma_1 = \alpha_1 \Sigma$, $\Sigma_2 = \alpha_2 \Sigma$, ... $\Sigma_C = \alpha_C \Sigma$.
  - **Fisher's linear discriminant analysis**: assume that $\Sigma_1 = \Sigma_2 = ... = \Sigma_C$.

Fisher's LDA

- Quadratic discriminant analysis classification was obtained above in (1):

$$g_j(x) = \log p(\omega_j) - \frac{1}{2}\log|\widehat{\Sigma}_j| - \frac{1}{2}(x-\widehat{\mu}_j)^T\widehat{\Sigma}_j^{-1}(x-\widehat{\mu}_j)$$

- Under assumption $\Sigma_1 = \Sigma_2 = ... = \Sigma_C$ we obtain:

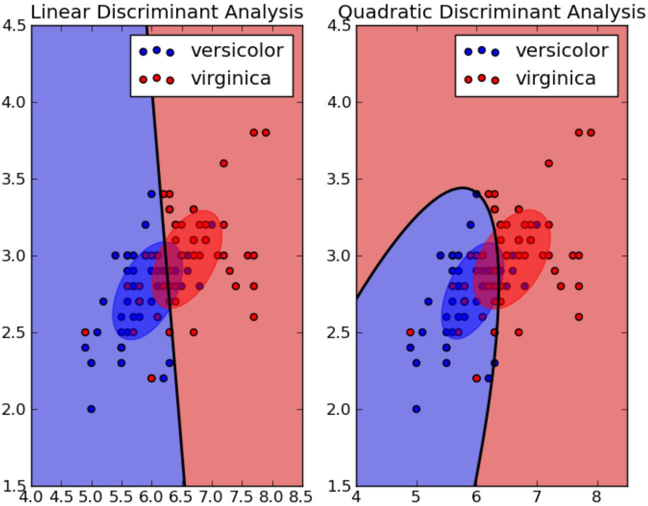
$$g_j(x) = \log p(\omega_j) - \frac{1}{2}\widehat{\mu}_j^T S_W^{-1}\widehat{\mu}_j + x^T S_W^{-1}\widehat{\mu}_j$$

- where estimator of common covariance matrix is given by pooled within class covariance matrix

$$S_W = \sum_{j=1}^{C}\frac{N_j}{N}\hat{\Sigma}_j$$

- Unbiased estimate of common covariance matrix is $\frac{N}{N-C}S_W$.

LDA vs. QDA


Linear Discriminant Analysis
Quadratic Discriminant Analysis
versicolor
virginica
4.5
4.0
3.5
3.0
2.5
2.0
1.5
4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
4 5 6 7 8

Addition

- For LDA: if $p(\omega_1) = p(\omega_2) = ... = p(\omega_C)$ and $\Sigma_1 = \Sigma_2 = ... = \Sigma_C = I$, then LDA reduces to **nearest mean classifier**.
- Regularized discriminant analysis - intermediate case between different, common and identity covariance matrices:

$$\widetilde{\Sigma}_j = \alpha\Sigma_j + \beta\Sigma + (1 - \alpha - \beta)I, \quad \alpha \geq 0, \beta \geq 0, \alpha + \beta \leq 1.$$

- α and β for RDA are selected using cross-validation.

Table of Contents

Mixture models

- Mixture models have the form:

$$p(x) = \sum_{j=1}^{g} \pi_j p(x; \theta_j)$$

- May be viewed as a two-step random process:
  1. generate cluster number j from discrete distribution

  $$j \sim Disc(\pi_1, \pi_2, \dots \pi_g)$$

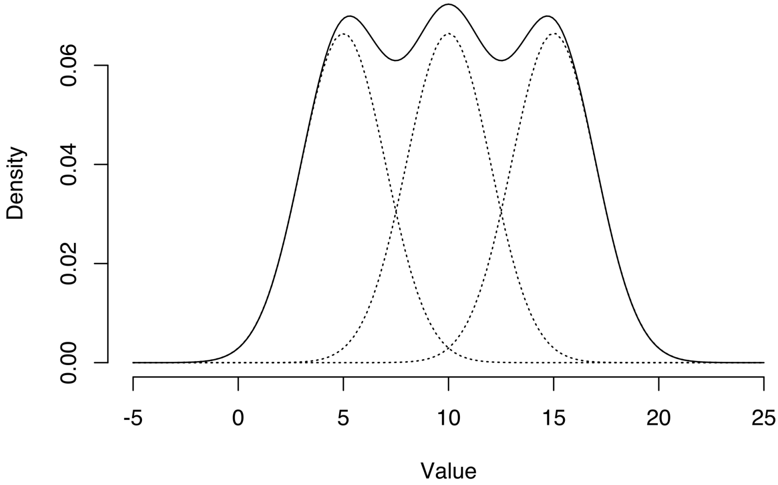
  2. generate observation x using density, representing given cluster $p(x; \theta_j)$

  $$x \sim p(x; \theta_j)$$

- Normal mixtures:

$$p(x) = \sum_{j=1}^{g} \pi_j N(x; \mu_j, \Sigma_j)$$

Normal mixture example



Likelihood maximization

- Direct optimization of likelihood is not possible:

$$L(\pi_1, ...\pi_g, \mu_1, ...\mu_g, \Sigma_1, ...\Sigma_g) = \prod_{i=1}^{N} \left\{ \sum_{j=1}^{g} \pi_j N(x; \mu_j, \Sigma_j) \right\}$$

- EM-algorithm is an iterative algorithm for likelihood maximization:

EM algorithm

- while parameters not converged:
  - for each sample $i = 1, 2, \ldots N$; for each cluster $j = 1, 2, \ldots g$:
    - recalculate cluster correspondences:

$$w_{ij} = \frac{\pi_j N(x_i; \mu_j, \Sigma_j)}{\sum_k \pi_k N(x_i; \mu_k, \Sigma_j)}$$

    - recalculate parameters of each cluster:

$$\hat{\pi}_j = \frac{1}{N} \sum_{i=1}^{N} w_{ij}, \quad \hat{\mu}_j = \frac{1}{\sum_{i=1}^{N} w_{ij}} \sum_{i=1}^{N} w_{ij} x_i$$

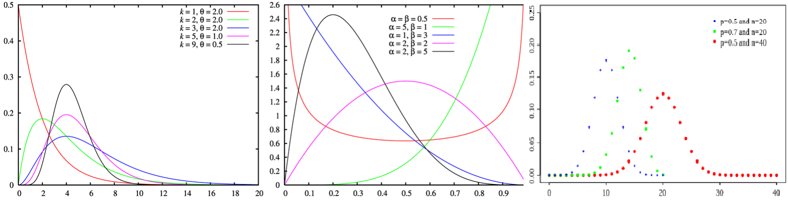
$$\hat{\Sigma}_j = \frac{1}{\sum_{i=1}^{N} w_{ij}} \sum_{i=1}^{N} w_{ij} \left(x_i - \hat{\mu}_j\right) \left(x_i - \hat{\mu}_j\right)^T$$

Application

- May converge to local optimum, depending on initial condition
  - starting EM from different initial conditions is advised
- Initial conditions may be obtained from k-means algorithm
- Initial covariance should be wide enough
- Degenerate optimums: $\hat{\mu}_j = x_k$, $\hat{\Sigma}_j \to \Theta \in \mathbb{R}^{DxD}$ (Θ is the zero matrix)
- $\hat{\Sigma}_j$ may become singular, so $N(x_i; \mu_j, \Sigma_j)$ will not be computable
  - $\tilde{\Sigma}_j \leftarrow \Sigma_j + \alpha I$
  - consider only diagonal Σ_j
  - consider only spherical $\Sigma_j = \alpha_j I$
  - assume common covariance matrix $\Sigma_1 = \Sigma_2 = ... = \Sigma_g$

Different distributions

- Other distributions may be taken instead of Gaussian:
  - **Gamma distribution** for non-negative random variables
  - **Beta distribution** for non-negative bounded random variables
  - **Polynomial distribution** for non-negative discrete random variables
  - etc.



Gamma, beta and binomial distributions