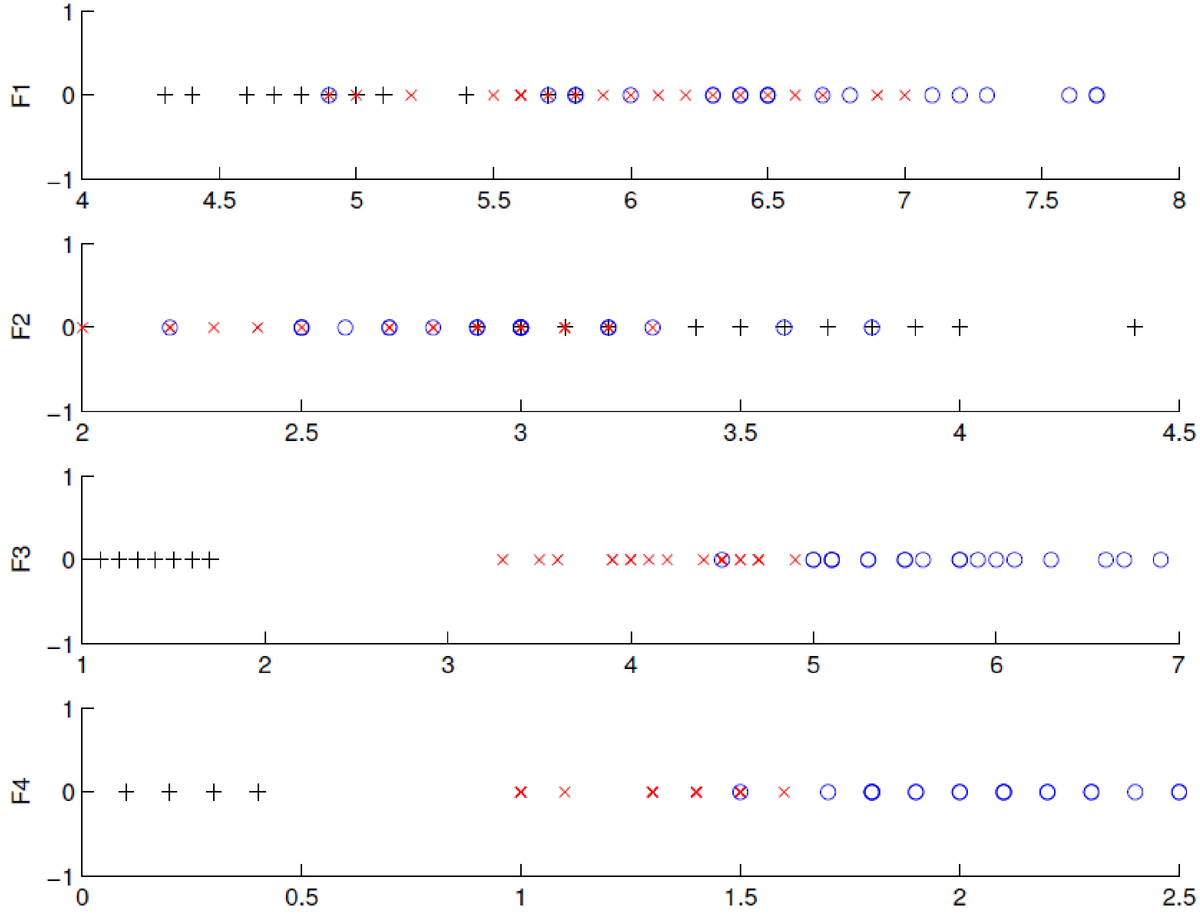
GAUSSIAN MIXTURE MODEL

Shingchern D. You

Why use GMM

- Sometimes data are not from one Gaussian



Why use GMM

- Therefore, we want to use mixture of Gaussian
- A Gaussian mixture model is a weighted sum of *m*-component Gaussian densities given by

$$f_y(\boldsymbol{x}_{(\boldsymbol{i})}|\theta) = \sum_j \alpha_j g(\boldsymbol{x}_{(\boldsymbol{i})}|\boldsymbol{\mu}_j, \Sigma_j)$$

where $\alpha_1 + \cdots + \alpha_m = 1$ and $\boldsymbol{x}_{(\boldsymbol{i})}$ is an outcome (observation) from a vector of RV $\boldsymbol{y}$ (or from iid $\boldsymbol{y}$)

- It can approximate any pdf (with sufficiently large *m*)

Why use GMM

- In general, μ_j (mean) is a column matrix and Σ_j (covariance) is a square matrix
- Want to find parameters from observations $\boldsymbol{x}_{(\boldsymbol{i})}$
- GMM can be used in two ways
  - Soft clustering (vs. k-means)
  - Classification (need *k* models for *k* classes)

Data Generation

- Assuming data are generated as follows
  - Pick one Gaussian randomly with a probability $P(g_j \text{ chosen}) = \alpha_j$
  - Multidimensional data point is generated from g_j
- How to estimate Gaussian parameters from data points

Parameter estimation

- Basic approach: expectation-maximization algorithm
- Expectation step: Assume that Gaussian parameters are known, want to estimate α_i
- Maximization step: Given α_i, update Gaussian parameters

EM algorithm

- We use the following simple example to illustrate how it works (Ref:
  http://www.cs.cmu.edu/~./awm/tutorials/gmm14.pdf or
  http://www.cs.nccu.edu.tw/~whliao/acv2008/08gmm.pdf)
- A more theoretical treatment can be found in the textbook or over Internet

EM algorithm

- We use three (univariate) Gaussian as an example

$$f_y(x_i|\theta) = p(x_i)$$
$$= \alpha_1 g_1(x_i) + \alpha_2 g_2(x_i) + \alpha_3 g_3(x_i)$$

- Suppose we have *n* observations $x_i, i = 1 \ldots n$
- Use maximum likelihood estimation to find $\theta = (\mu_1, \mu_{2,} \mu_3, \sigma_1^2, \sigma_2^2, \sigma_3^2)$
- Let $J(\theta) = \ln \prod_i p(x_i)$

$$= \sum_i \ln[\alpha_1 g_1(x_i) + \alpha_2 g_2(x_i) + \alpha_3 g_3(x_i)]$$

EM algorithm (M step)

- To find optimal values for θ, we use derivatives

$$\frac{\partial}{\partial \mu_j} J(\theta) = 0$$

$$\frac{\partial}{\partial \sigma_j} J(\theta) = 0$$

EM algorithm (M step)

- With some computations, we have (M step)

$$\mu_j = \frac{\sum_{i=0}^{n} \beta_j(x_i)\, x_i}{\sum_{i=0}^{n} \beta_j(x_i)}$$

$$\sigma_j^2 = \frac{\sum_{i=0}^{n} \beta_j(x_i)(x_i - \mu_j)^2}{\sum_{i=0}^{n} \beta_j(x_i)}$$

where $\beta_j(x) = P(j|x) = \frac{\alpha_j g_j(x)}{\sum_{k=1}^{3} \alpha_k g_k(x)}$

EM algorithm (E step)

- We also need to estimate α_i by letting (E step)

$$\frac{\partial}{\partial \alpha_j} J(\theta) = 0$$

However, we have one constraint: $\alpha_1 + \cdots + \alpha_3 = 1$

- This can be solved with Lagrange multipliers
- Solving it, we have $\alpha_j = \frac{1}{n} \sum_{i=0}^{n} \beta_j(x_i)$
- It can be proved that likelihood never decreases for each iteration (going through E and M steps)

Using EM algorithm

- Set some initial values for α_j (if no a priori knowledge, set $\alpha_j = \frac{1}{m}$ (m: # of mixtures)
- Set some initial values for $(\mu_1, \mu_{2,}\mu_3, \sigma_1^2, \sigma_2^2, \sigma_3^2)$
- If needed, we can use k-means clustering algorithm to find $\mu_1, \mu_{2,}\mu_3$
- Iterate E and M steps until the change of parameters are very small (or increase of likelihood function is very small)

Using EM algorithm

- GMM for classification
- Step 1: Train one GMM per class
- Step 2: Put the unknown input x into likelihood

Function for each model

- Step 3: Classify x belonging to the class with highest likelihood