Introduction to Machine Learning

Ranga Raju Vatsavai, Ph.D.
Chancellors Faculty Excellence Associate Professor in Geospatial Analytics
Department of Computer Science, North Carolina State University (NCSU)

Feb. 25-27, 2019

Semi-supervised Learning

- Ground-truth is costly and time consuming
  - Empirical studies show that we need at least (10-30) x dimensions number of samples per class
- Unlabeled samples are plenty and cheap
- Can we combine small number of labeled samples and large number of unlabeled sample to improve learning model?
  - Yes: semi-supervised learning

Self-training

- Key assumption
  - One's own high confidence predictions are correct
- Algorithm
  1. Train f from D_l = (**X**,Y)
  2. Use f to predict on D_{ul} = (**x**, ?)
  3. Add (**x**, f(**x**)) to labeled data D_l
     - Variations: All, most confident, weighted?
  4. Repeat

Self-training

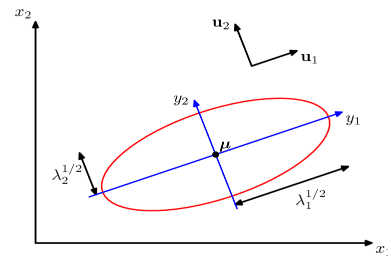
- Advantages
  - Simplest semi-supervised learning
  - Easy to implement (wrapper around existing algorithms)
- Disadvantages
  - Early mistakes may lead to degrade in performance

Probabilistic Approach

- Assume data is generated by a mixture model
  - E.g., Gaussian Mixture Model
- D = Dl + Dul
- How do you estimate parameters? MLE don't work with missing labels
- Solution
  - Expectation Maximization (EM) algorithm

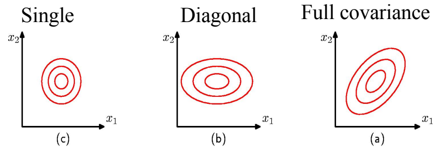
Multivariate Gaussian

- Bivariate covariance matrix representation



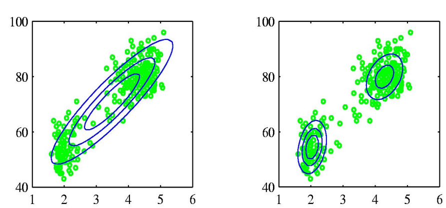
Multivariate Gaussian

- Σ plays key role, but accurate estimation requires large number samples (10-30 x dim; per class)
- However, Σ can be simplified



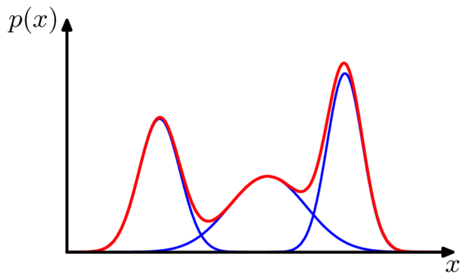
$$\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 \end{pmatrix}$$

What If The Data Is Multimodal



- Real world data are rarely unimodal
- Many times we don't know the labels for all components

How to model such data?

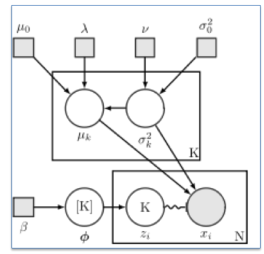


- Mixtures of Gaussians
- Mixture models in general

Mixture Models

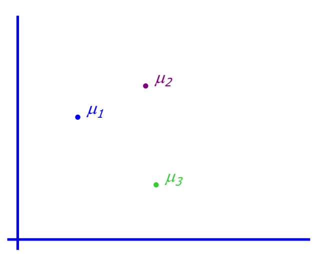
- Mixture model is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs.

- K = # components
- N = # observations
- θ = parameter
- Φ = mixture weight
- z_i = component of observation i
- x_i = i^{th} observation



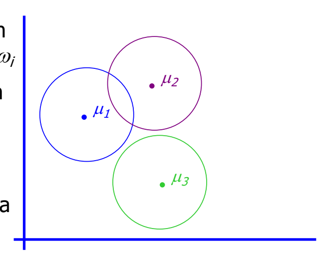
The GMM assumption

- There are k components. The i' th component is called ω_i
- Component ω_i has an associated mean vector μ_i



The GMM assumption

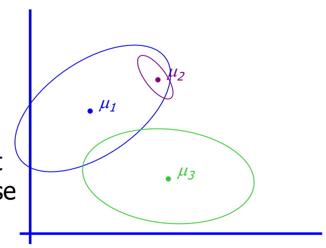
- There are k components. The i' th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$



The General GMM assumption

Assume that each data point is generated according to the following recipe:

1. Pick a component at random. Choose component i with probability $P(\omega_i)$.
2. Data point ~ $N(\mu_i, \Sigma_i)$



GMM

- GMM (K=M) $p(x|\theta) = \sum_{i=1}^{M} \alpha_i p_i(x|\theta_i)$

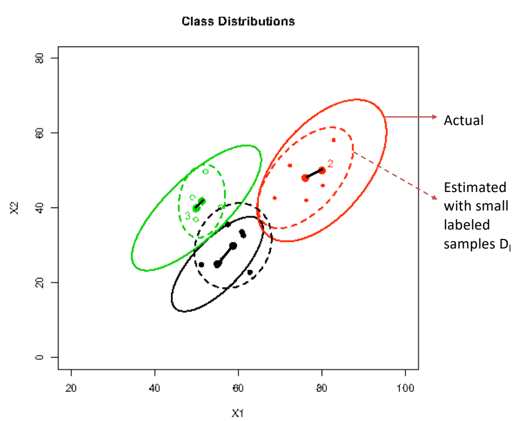
  where $\theta = (\alpha_1, ..., \alpha_M; \theta_1, ...\theta_M)$

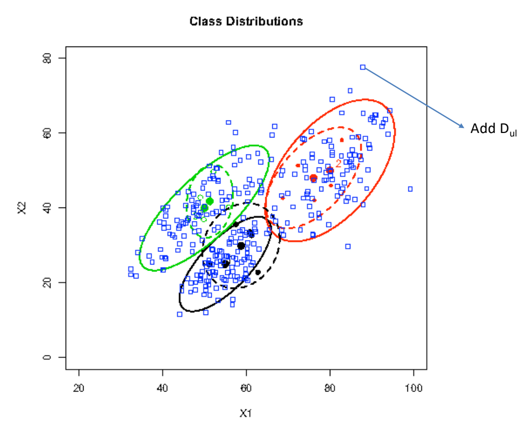
  such that $\sum_1^M \alpha_i = 1, \quad 0 < \alpha_i < 1$ and

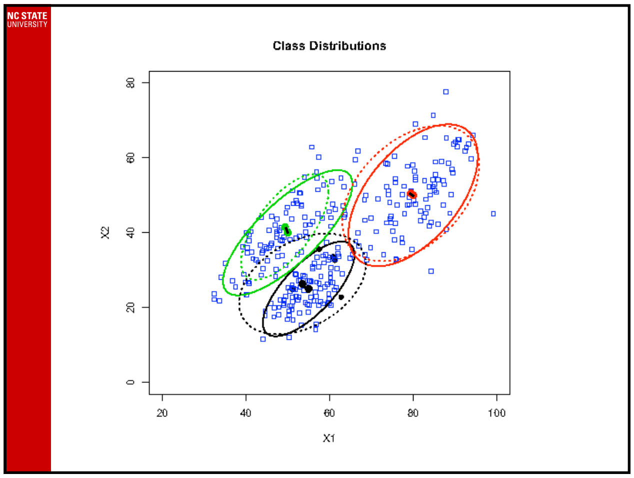
  p_i pdf parameterized by θ_i

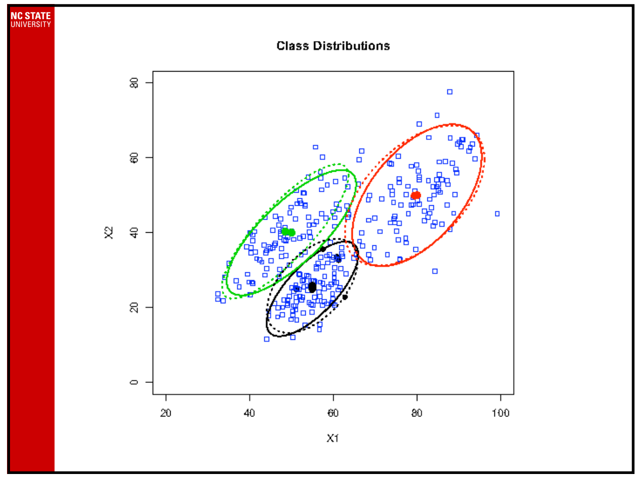
- Maximize

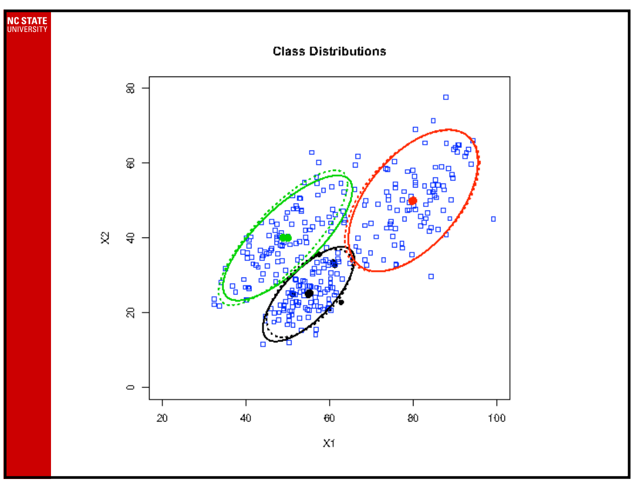
$$L(\theta) - L(\theta_i) = \ln \frac{\sum_z p(x|z,\theta)p(z|\theta)}{p(x|\theta_i)}$$

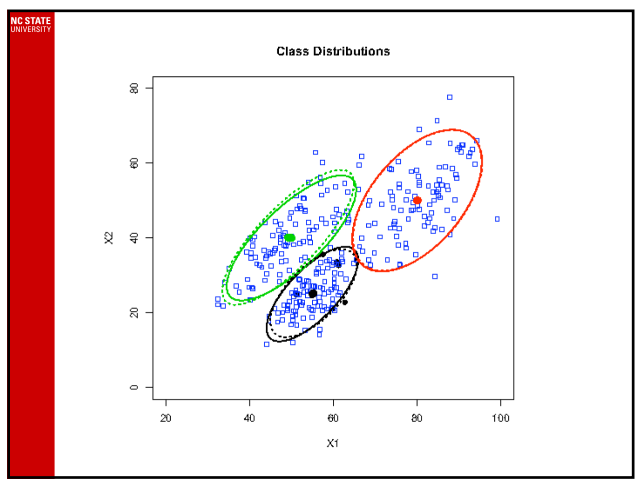





NC STATE UNIVERSITY
Class Distributions
X1
X2


NC STATE UNIVERSITY
Class Distributions
X1
X2


NC STATE UNIVERSITY
Class Distributions
X1
X2


NC STATE UNIVERSITY
Class Distributions
X1
X2

NC STATE UNIVERSITY

Semi-supervised Learning

EM to estimate GMM parameters

- E-Step

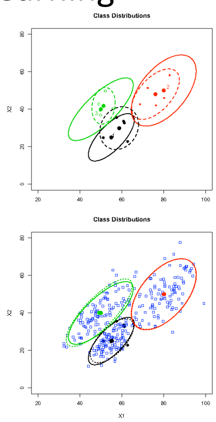
$$e_{ij} = \frac{\left|\hat{\Sigma}_j^k\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(x_i - \hat{\mu}_j^k\right)^T \hat{\Sigma}_j^{-1,k}\left(x_i - \hat{\mu}_j^k\right)\right\}}{\sum_{l=1}^{M}\left|\hat{\Sigma}_l^k\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(x_i - \hat{\mu}_l^k\right)^T \hat{\Sigma}_l^{-1,k}\left(x_i - \hat{\mu}_l^k\right)\right\}}$$

- M-Step

$$\alpha_j = \frac{\sum_{i=1}^{N} e_{ij}}{N}, \qquad \hat{\mu}_j^{k+1} = \frac{\sum_{i=1}^{N} e_{ij} x_i}{\sum_{i=1}^{N} e_{ij}},$$

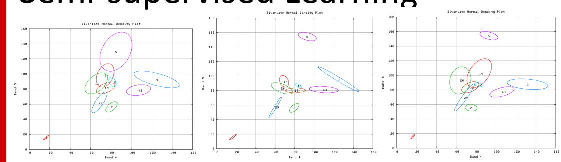
$$\text{and} \quad \hat{\Sigma}_j^{k+1} = \frac{\sum_{i=1}^{N} e_{ij}\left(x_i - \hat{\mu}_j^{k+1}\right)\left(x_i - \hat{\mu}_j^{k+1}\right)^T}{\sum_{i=1}^{N} e_{ij}}$$

i^{th} data vector, j^{th} class



NC STATE UNIVERSITY

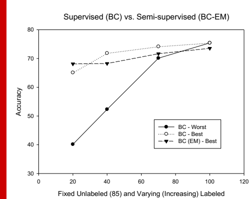
Semi-supervised Learning



10 Classes, 100 Training Samples (10-30) x No of dimensions / class

Small Subset of 20 Training Samples

20 labeled + 80 unlabeled samples



Ranga Raju Vatsavai, Shashi Shekhar, Thomas E. Burk: A Semi-Supervised Learning Method for Remote Sensing Data Mining. ICTAI 2005: 207-211