TTT4275 Lecture 2
Spring 2018

Faglærer: Magne Hallstein Johnsen,

Institutt for elektronikk og telekommunikasjon, NTNU

Lecture content

- Designing and training parametric BDR classifiers
  - Different design and training issues
  - The likelihood concept and Maximum likelihood (ML) training
  - Maximum A Posteriori (MAP) training
  - A short intro to Expectation-Maximation (E-M) algorithm applied to training of Gaussian mixture models (GMM)

Different design and training issues for BDR classifiers

- Assuming continuous input x
  - Is x a (static) vector or a vector sequence (temporal information)
  - If x is vector how is x distributed (GMM?)
  - If x is a sequence we also have to model (by HMM?) the temporal dependency.
  - In this course we assume x is a $\underline{\text{v}}$ector
- Training issues
  - We need a training set $X = \{x_k \; k = 1, N\})$
  - Parameters Θ must be estimated
  - Usually supervised training (training set X is class labeled) is applied
  - Training set size and representativeness are important to fullfil !

The likelihood concept and ML training : part 1

- Defining log likelihood for an unknown parameter set Θ
  - Given/measured a labeled training set X) :
  - $LL(\Theta) = log[p(X/\Theta)] = log[\prod_{k=1}^{N} p(x_k/\Theta)] = \sum_{k=1}^{N} log[p(x_k/\Theta)]$
  - Assumes independent training observations
  - Same principle as curve (i.e. model) fitting!

- ML training :
  - Find Θ which maximizes $LL(\Theta)$ given X
  - In some cases this is an intrinsic optimization problem (iterative two-step algorithms, like the E-M, are needed)

The likelihood concept and ML training : part 2

- The classical Gaussian vector case $p(x/\omega_i) = N(\mu_i, \Sigma_i)$ $i = 1, C$
- Parameters for each class $\omega_i \Rightarrow \Theta_i = \{\mu_i, \Sigma_i\}$ is found separately !
- Omitting class index for briefity and assume $X \in \omega$
- $LL(\Theta) = Konst - 0.5 N log(|\Sigma|) - 0.5 \sum_{k=1}^{N} (x_k - \mu)^T \Sigma^{-1} (x_k - \mu)$
- ML $\Rightarrow \nabla_{\Theta} LL(\Theta) = 0$
  - ML $\Rightarrow \nabla_{\mu} LL(\Theta) = \sum_{k=1}^{N} \Sigma^{-1}(x_k - \mu) = 0 \Rightarrow$
  - $\mu_{ML} = (1/N) \sum_{k=1}^{N} x_k$ (sample mean)
  - A similar procedure gives :
  - $\Sigma_{ML} = (1/N) \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T$ (sample covariance)

Maximum A Posteriori - MAP training : part 1

- ML assumes Θ is unknown, deterministic
- How to include some knowledge about Θ? A popular strategy is to use a statistical approach $\Rightarrow p(\Theta)$ (prior density - must be chosen)
- MAP $\Rightarrow \max_{\Theta} log[P(\Theta/X)] \equiv \max_{\Theta} log(p(X/\Theta)p(\Theta))$
- $\Rightarrow log[P(\Theta/X)] = log[p(\Theta)] + \sum_{k=1}^{N} log[p(x_k/\Theta)]$
- Explicit solutions are dependent of choosing a "manageable" (in a mathematical sense) prior distribution.
- In the Gaussian vector case $p(x/\omega) = N(\mu, \Sigma)$ this restriction leads to the following choices
  - $p(\mu) = N(\mu_0, \Sigma_0)$ (see next slide)
  - while the Wishart distribution has to be chosen for $p(\Sigma)$
- Several methods have been developed with respect to choosing good prior (hyper)parameters

Maximum A Posteriori - MAP training : part 2

- Assuming the Gaussian vector case and that only μ must be estimated.
- This means that the prior $\{\mu_0, \Sigma_0\}$ is chosen and the sample covariance estimate $\Sigma = \Sigma_{ML}$ is good enough
- $\nabla_\mu log[p(\Theta/X)] = 0 \;\Rightarrow\; \mu_{MAP} = (N\Sigma^{-1} + \Sigma_0^{-1})^{-1}(N\Sigma^{-1}\mu_{ML} + \Sigma_0^{-1}\mu_0)$
- Here μ_{ML} is the sample mean.
- Note : a) $N \to \infty \;\Rightarrow\; \mu_{MAP} \to \mu_{ML}$ b) $N \to 0 \;\Rightarrow\; \mu_{MAP} \to \mu_0$
- For the special case when x is a scalar we get :

$$\mu = \frac{\sigma^2\mu_0 + N\sigma_0^2\mu_{ML}}{\sigma^2 + N\sigma_0^2} \tag{1}$$

Gaussian mixture modeling and the EM algorithm

- In most cases the true (but unknown) class distribution deviates from a single Gaussian

- However, a weighted sum of L Gaussians can approximate any continuous distribution

- GMM : $p(x/\omega) = p(x/\Theta, P_{all}) = \sum_{j=1}^{L} p(x/\Theta, j)P_j = \sum_{j=1}^{L} P_j N(\mu_j, \Sigma_j)$ where $\Theta = [\mu_j, \Sigma_j]$ and $P_{all} = [P_j \;\; j = 1, L]$

- We assume x is drawn with probability P_j from Gaussian number j. Note $P_j = c_j$, the weight for mixture j.

- If j_k was known for every x_k $k = 1, N$ in the training set, we could find the parameters for each Gaussian separately (by ML or MAP)

- However, the j_k's are unknown; i.e. we say that the data set $y_k \triangleq \{x_k, j_k\}$ are incomplete (an insintric problem).

- Note : This GMM problem is an example of so called clustering (lectured later in this course). This also includes how to decide upon the optimal number of gaussians/clusters.