Bayes Decision Rule and Naïve Bayes Classifier

Le Song

Machine Learning I
CSE 6740, Fall 2013
Gaussian Mixture model

- A density model $p(X)$ may be multi-modal: model it as a mixture of uni-modal distributions (e.g. Gaussians)

- Consider a mixture of $K$ Gaussians
  
  $p(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X | \mu_k, \Sigma_k)$

- Learn $\pi_k \in (0,1), \mu_k, \Sigma_k$;

- Mixing proportion

- Mixture Component
Imagine generating your data ...

- For each data point \( x^i \):
  - Randomly choose a mixture component, \( z^i = \{1, 2, \ldots, K\} \), with probability \( p(z^i | \pi) = \pi_{z^i} \)
  - Then sample the actual value of \( x^i \) from a Gaussian distribution \( p(x^i | z^i) = \mathcal{N}(x | \mu_{z^i}, \Sigma_{z^i}) \)

- Joint distribution over \( x \) and \( z \)
  \[
p(x, z) = p(z)p(x | z) = \pi_z \mathcal{N}(x | \mu_z, \Sigma_z)
  \]

- Marginal distribution \( p(x) \)
  \[
p(x) = \sum_{z=1}^{K} p(x, z) = \sum_{z=1}^{K} \pi_z \mathcal{N}(x | \mu_z, \Sigma_z)
  \]
Bayes rule

\[ P(z|x) = \frac{P(x|z)p(z)}{P(x)} = \frac{P(x,z)}{\sum_z P(x,z)} \]

Prior: \( p(z) = \pi_z \)
Likelihood: \( p(x|z) = \mathcal{N}(x|\mu_z, \Sigma_z) \)
Posterior: \( p(z|x) = \frac{\pi_z \mathcal{N}(x|\mu_z, \Sigma_z)}{\sum_z \pi_z \mathcal{N}(x|\mu_z, \Sigma_z)} \)
Details of EM

- We intend to learn the parameters that maximizes the log-likelihood of the data

\[
    l(\theta; D) = \log \prod_{i=1}^{m} \left( \sum_{z_i^1}^{K} p(x^i, z^i|\theta) \right)
\]

- Expectation step (E-step): What do we take expectation over?

\[
    f(\theta) = E_{q(z^1, z^2, ..., z^m)} \left[ \log \prod_{i=1}^{m} p(x^i, z^i|\theta) \right]
\]

- Maximization step (M-step): how to maximize?

\[
    \theta^{t+1} = \text{argmax}_\theta f(\theta)
\]
E-step: what is $q(z^1, z^2, \ldots, z^m)$

- $q(z^1, z^2, \ldots, z^m)$: posterior distribution of the latent variables

$$q(z^1, z^2, \ldots, z^m) = \prod_{i}^{m} p(z^i | x^i, \theta^t)$$

- For each data point $x^i$, compute $p(z^i = k | x^i)$ for each $k$

$$\tau_k^i = p(z^i = k | x^i)$$

$$\tau_k^i = \frac{\pi_k \mathcal{N}(x^i | \mu_k, \Sigma_k)}{\sum_k \pi_k \mathcal{N}(x^i | \mu_k, \Sigma_k)}$$
E-step: compute the expectation

\[ f(\theta) = E_{q(z^1, z^2, \ldots, z^m)} \left[ \log \prod_{i=1}^{m} p(x^i, z^i | \theta) \right] \]

\[ = \sum_{i=1}^{m} E_{p(z^i | x^i, \theta^t)} [\log p(x^i, z^i | \theta)] \]

In the case of Gaussian mixture:

\[ f(\theta) = \sum_{i=1}^{m} E_{p(z^i | x^i, \theta^t)} \left[ \log \pi_z^i - (x^i - \mu_{z^i})^T \Sigma_z (x^i - \mu_{z^i}) + \log \Sigma_z + c \right] \]

\[ = \sum_{i=1}^{m} \sum_{k=1}^{K} \tau_k^i \left[ \log \pi_k - (x^i - \mu_k)^T \Sigma_k (x^i - \mu_k) + \log \Sigma_k + c \right] \]
M-step: maximize $f(\theta)$

$$f(\theta) = \sum_{i=1}^{m} \sum_{k=1}^{K} \tau_i^k \left[ \log \pi_k - (x^i - \mu_k)^T \Sigma_k (x^i - \mu_k) + \log \Sigma_k + c \right]$$

For instance, we want to find $\pi_k$, and $\sum_{i=1}^{K} \pi_k = 1$

- Form Lagrangian

$$L = \sum_{i=1}^{m} \sum_{k=1}^{K} \tau_i^k [\log \pi_k + \text{other terms}] + \lambda (1 - \sum_{i=1}^{K} \pi_k)$$

- Take partial derivative and set to 0

$$\frac{\partial L}{\partial \pi_k} = \sum_{i=1}^{m} \frac{\tau_i^k}{\pi_k} - \lambda = 0$$

$$\Rightarrow \pi_k = \frac{1}{\lambda} \sum_{i=1}^{m} \tau_i^k$$

$$\Rightarrow \lambda = m$$
EM graphically

\[ f(\theta) \]

\[ l(\theta^t; D) \]

M – step \( \theta^{t+1} \)

E – step

\[ l(\theta; D) \]
Experiments with mixture of 3 Gaussians
Why do we need density estimation?

- Learn the “shape” of the data cloud

- Assess the likelihood of seeing a particular data point
  - Is this a typical data point? (high density value)
  - Is this an abnormal data point / outlier? (low density value)

- Building block for more sophisticated learning algorithms
  - Classification, regression, graphical models ...
Outlier detection

- Declare data points in low density region as outliers
  - First fit density $p(x)$, and then set a threshold $t$
  - Data points in region with density small than $t$, i.e., $p(x) \leq t$
Geometric method for outlier detector

- Rather than first fit density and then threshold, directly find the boundary between inlier and outlier
Minimum enclosing ball

- Find the smallest ball such that it include all data points.

- Parameterized with a center $c$ and radius $r$

- Optimization problem (quadratically constrained quadratic programming)

\[
\begin{align*}
\min_{r,c} & \quad r^2 \\
\text{s.t.} & \quad (x^i - c)^\top (x^i - c) \leq r^2, \quad \forall i = 1, \ldots, m
\end{align*}
\]
Find center \( c \) and radius \( r \)

- Lagrangian of the problem:

\[
L = r^2 + \sum_{i=1}^{m} \alpha_i \left( (x^i - c)^T (x^i - c) - r^2 \right), \alpha_i \geq 0
\]

- Set partial derivative of \( L \) to 0

\[
\frac{\partial L}{\partial r} = 2r - 2 \sum_{i=1}^{m} \alpha_i r = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i = 1
\]

\[
\frac{\partial L}{\partial c} = \sum_{i=1}^{m} \alpha_i (-2x^i + 2c) = 0 \Rightarrow c = \sum_{i=1}^{m} \alpha_i x^i
\]

c and r not in explicit form
Manipulate the Lagrangian

\[ L = r^2 + \sum_{i=1}^{m} \alpha_i \left( (x^i - c)^\top (x^i - c) - r^2 \right) \]

\[ L = r^2 \left( 1 - \sum_{i=1}^{m} \alpha_i \right) + \sum_{i} \alpha_i x^i \top x^i - 2c^\top \left( \sum_{i} \alpha_i x^i \right) \]

\[ + \left( \sum_{i=1}^{m} \alpha_i \right) c^\top c \]

\[ L = \sum_{i=1}^{m} \alpha_i x^i \top x^i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j x^i \top x^j \]
Express Lagrangian $L$ in $\alpha$

$$\max_{\alpha} \quad L(\alpha) = \sum_{i=1}^{m} \alpha_i x_i^\top x_i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j x_i^\top x_j$$

s.t. $\sum_{i=1}^{m} \alpha_i = 1$, $\alpha_i \geq 0$

A quadratic programming problem $L = b^\top \alpha - \alpha^\top A \alpha$ with linear constraints

\[ A_{ij} = x_i^\top x_j \]  
\[ b_i = x_i^\top x_i \]  
\[ \alpha \]
Experiment with minimum enclosing ball
Classification

- Represent the data

- A label is provided for each data point, e.g., $y \in \{-1, +1\}$

- Classifier
Outline

- What is the theoretically the best classifier?
- Bayesian decision rule for minimum error.
- What is the difficulty of achieve the minimum error?
- What do we do in practice?
Decision-making: divide high-dimensional space

- Distributions of sample from normal (positive class) and abnormal (negative class) tissues
In classification case, we are given the label for each data point.

Class conditional distribution: $P(x|y = 1), P(x|y = -1)$ (how to compute?)

Class prior: $P(y = 1), P(y = -1)$ (how to compute?)
How to come up with decision boundary

- Given class conditional distribution:  \( P(x|y = 1), P(x|y = -1) \), and class prior:  \( P(y = 1), P(y = -1) \)

\[
P(x|y = 1) = \mathcal{N}(x; \mu_1, \Sigma_1)
\]

\[
P(x|y = -1) = \mathcal{N}(x; \mu_2, \Sigma_2)
\]
Use Bayes rule

\[ P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{\sum_z P(x,y)} \]

Prior: \( P(y) \)
Likelihood (class conditional distribution): \( p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y) \)

Posterior: \( P(y|x) = \frac{P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}{\sum_y P(y)\mathcal{N}(x|\mu_y, \Sigma_y)} \)
Bayes Decision Rule

- Learning: prior: $p(y)$, class conditional distribution: $p(x|y)$

- The poster probability of a test point

$$q_i(x) := P(y = i|x) = \frac{P(x|y)P(y)}{P(x)}$$

- Bayes decision rule:
  - If $q_i(x) > q_j(x)$, then $y = i$, otherwise $y = j$

- Alternatively:
  - If ratio $l(x) = \frac{P(x|y=i)}{P(x|y=j)} > \frac{P(y=j)}{P(y=i)}$, then $y = i$, otherwise $y = j$
  - Or look at the log-likelihood ratio $h(x) = -\ln \frac{q_i(x)}{q_j(x)}$
Example: Gaussian class conditional distribution

- Depending on the Gaussian distributions, the decision boundary can be very different

\[
\text{Decision boundary: } h(x) = -\ln \frac{q_i(x)}{q_j(x)} = 0
\]
Is Bayesian decision rule optimal?

- Optimal with respect to what criterion?

- Bayes error: expected probability of a sample being incorrectly classified

  - Given a sample, we compute posterior distribution \( q_1(x) := P(y = 1|x) \) and \( q_2(x) := P(y = 2|x) \), the probability of error:

    \[
    r(x) = \min\{q_1(x), q_2(x)\}
    \]

  - Bayes error

    \[
    E_{p(x)}[r(x)] = \int r(x)p(x)dx
    \]
Bayes error

The Bayes error:

\[ \epsilon = E_{p(x)}[r(x)] = \int r(x)p(x)dx \]

\[ = \int \min\{q_1(x), q_2(x)\} p(x)dx \]

\[ = \int \min \left\{ \frac{P(y = 1)p(x|y = 1)}{p(x)}, \frac{p(y = 2)p(x|y = 2)}{p(x)} \right\} p(x)dx \]

\[ = \int \min\{P(y = 1)p(x|y = 1), P(y = 2)p(x|y = 2)\} dx \]
Bayes error II

\[ \epsilon = \int \min\{P(y = 1)p(x|y = 1), P(y = 2)p(x|y = 2)\} \, dx \]

\[ = \int_{L_1} P(y = 1)p(x|y = 1) \, dx + \int_{L_2} P(y = 2)p(x|y = 2) \, dx \]
More on Bayes error

- Bayes error is the lower bound of probability of classification error

- Bayes decision rule is the theoretically best classifier that minimize probability of classification error

- However, computing Bayes error or Bayes decision rule is in general a very complex problem. Why?

  - Need density estimation

  - Need to do integral, eg. $\int_{L_1} P(y = 1)p(x|y = 1) \, dx$
What do people do in practice?

- Use simplifying assumption for $p(x|y = 1)$
  - Assume $p(x|y = 1)$ is Gaussian
  - Assume $p(x|y = 1)$ is fully factorized

- Rather than model density of $p(x|y = 1)$, directly go for the decision boundary $h(x) = -\ln \frac{q_i(x)}{q_j(x)}$
  - Logistic regression
  - Support vector machine
  - Neural networks
Naïve Bayes Classifier

- Use Bayes decision rule for classification

\[ P(y|x) = \frac{P(x|y)P(y)}{P(x)} \]

- But assume \( p(x|y = 1) \) is fully factorized

\[ p(x|y = 1) = \prod_{i=1}^{d} p(x_i|y = 1) \]

- Or the variables corresponding to each dimension of the data are independent given the label