Machine Learning

Week 2: Pattern Classification

Mahesan Niranjan

School of Electronics and Computer Science
University of Southampton

Autumn Semester 2016/17

Overview (Week 2)

- Review of what we learnt in Lab One
  - Multivariate Gaussian
  - Drawing samples from $\mathcal{N}\left(\boldsymbol{m}, \boldsymbol{C}\right)$
  - Principal directions
- Introduction to Bayesian Decision Theory
- Bayes' Classifier for Simple Gaussian Distributions
- Simple Classifiers
  - Distance to mean classifier
  - Nearest Neighbour classifier
  - Linear classifier (more on this later)
  - Perceptron (formal setting later)
- What will we learn in Lab Two?

Bayesian Decision Theory

- Classes: $\omega_i, \; i = 1, ..., K$
- Prior Probabilities: $P[\omega_1], ..., P[\omega_K]$; $P[\omega_i] \geq 0, \;\; \sum_{i=1}^{C} P[\omega_i] \; = \; 1$
- Likelihoods (class conditional probabilities): $p(\boldsymbol{x}|\omega_i), \; i = 1, .., K$
- Posterior Probability: $P[\omega_j \,|\, \boldsymbol{x}]$

$$P[\omega_j \,|\, \boldsymbol{x}] \;=\; \frac{p(\boldsymbol{x} \,|\, \omega_j) \;\; P[\omega_j]}{\sum_{i=1}^{K} p(\boldsymbol{x} \,|\, \omega_i) \; P[\omega_i]}$$

- From prior knowledge: $P[\omega_i]$; From traing data: $p(\boldsymbol{x}|\omega_i)$
- Decision rule: Assign $\boldsymbol{x}$ to the class that maximizes posterior probability.
- The denominator is a constant; *i.e.* does not depend on ω_j
- Hence the decision rule becomes:

$$\boldsymbol{x} \in \max_j p(\boldsymbol{x} \,|\, \omega_j) \;\; P[\omega_j]$$

Bayes' Classifier for Gaussian Densities

Make assumptions, cancel common terms when making comparisons...

- Decision rule from: $p(\boldsymbol{x} \,|\, \omega_j) \;\; P[\omega_j]$
- Assume the two classes are Gaussian distributed with distinct means and identical covariance matrices $p(\boldsymbol{x} \,|\, \omega_j) \; = \; \mathcal{N}(\boldsymbol{m}_j, \, \boldsymbol{C})$
- Substitute into Bayes' classifier decision rule

$$\begin{aligned} P[\omega_1|\boldsymbol{x}] \quad &\lessgtr \quad P[\omega_2|\boldsymbol{x}] \\ p(\boldsymbol{x}|\omega_1) \;\; P[\omega_1] \quad &\lessgtr \quad p(\boldsymbol{x}|\omega_2) \;\; P[\omega_2] \end{aligned}$$

$$\frac{1}{(2\pi)^{p/2}(\det(C))^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_1)^t \boldsymbol{C}^{-1}(\boldsymbol{x} - \boldsymbol{m}_1)\right\} P[\omega_1] \quad \lessgtr$$
$$\frac{1}{(2\pi)^{p/2}(\det(C))^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_2)^t \boldsymbol{C}^{-1}(\boldsymbol{x} - \boldsymbol{m}_2)\right\} P[\omega_2]$$

Bayes' classifier for simple densities (cont'd)

Distinct Means; Equal, isotropic covariance matrix

- Suppose the densities are isotropic and priors are equal *i.e.* $\boldsymbol{C} = \sigma^2 \boldsymbol{I}$ and $P[\omega_1] = P[\omega_2]$
- The comparison simplifies to (see algebra on board):

$$\begin{aligned} (\boldsymbol{x} - \boldsymbol{m}_1)^t (\boldsymbol{x} - \boldsymbol{m}_1) & \lessgtr & (\boldsymbol{x} - \boldsymbol{m}_2)^t (\boldsymbol{x} - \boldsymbol{m}_2) \\ |\boldsymbol{x} - \boldsymbol{m}_1| & \lessgtr & |\boldsymbol{x} - \boldsymbol{m}_2| \end{aligned}$$

- The above is a simple *distance to mean* classifier
- Under the above simplistic assumptions, we only need to store one template per class (the means)!

Bayes' classifier for simple densities (cont'd)

Distinct Means; Common covariance matrix (but not isotropic)

- Cancel common terms and take log

$$(\boldsymbol{x} - \boldsymbol{m_1})^t \, \boldsymbol{C}^{-1} \, (\boldsymbol{x} - \boldsymbol{m_1}) \lessgtr (\boldsymbol{x} - \boldsymbol{m_2})^t \, \boldsymbol{C}^{-1} \, (\boldsymbol{x} - \boldsymbol{m_2}) - \log \left\{ \frac{P[\omega_1]}{P[\omega_2} \right\}$$

- Also simplifies to a $\boxed{\text{linear classifier}}$!

$$\boldsymbol{w}^t \, \boldsymbol{x} + b \lessgtr 0$$

$$\begin{aligned} \boldsymbol{w} & = & 2\boldsymbol{C}^{-1} \, (\boldsymbol{m}_2 - \boldsymbol{m}_1) \\ b & = & \left(\boldsymbol{m}_1^t \boldsymbol{C}^{-1} \boldsymbol{m}_1 \; - \; \boldsymbol{m}_2^t \boldsymbol{C}^{-1} \boldsymbol{m}_2 \right) - \log \left\{ \frac{P[\omega_1]}{P[\omega_2]} \right\} \end{aligned}$$

- Also a distance to template classifier, where the distance is

$$(\boldsymbol{x} - \boldsymbol{m_1})^t \, \boldsymbol{C}^{-1} \, (\boldsymbol{x} - \boldsymbol{m_1})$$

Known as Mahalanobis distance

Implementing a linear classifier: Perceptron

Error correcting learning

Linear classifier

$$\boldsymbol{w}^t\,\boldsymbol{x} + b \lessgtr 0$$

Expand dimensions: $\boldsymbol{a} = [\boldsymbol{w}^t\; b]^t$ and $\boldsymbol{y} = [\boldsymbol{x}^t\; 1]^t$

$$\boldsymbol{a}^t\,\boldsymbol{y} \lessgtr 0$$

```
random guess of the weights
repeat
    select data at random
    if not correctly classified
       update weights
until (all data correctly classified)
```

Update:

$$\boldsymbol{a}^{(k+1)} = \boldsymbol{a}^{(k)} + \eta\,\boldsymbol{y}^{(k)}$$

Lab 2

1. Some plotting
2. Bayes' optimal class boundary
3. Implement your own perceptron algorithm