Lecture 09

Linear Models for Classification
CS542—Contents 09

- Classification
  - Classification the Bayesian way
  - Discriminant functions
  - Linear discriminant function for two classes
  - Linear discriminant function for multiple classes
  - Least squares discrimination

- Handout:
  - Class Notes

- Reading Assignment for Next Class
  - Bishop Ch.4
Classification: The Bayesian Way

- Bayesian classification is optimal if the probability distributions are known

- Example:
  - Classify cars on the USC campus whether the cost more or less than $50k
    - Given:
      - One input feature $x$: “The height of the car”
      - Two classes:
        - $C_1$: price > $50k
        - $C_2$: price <= $50k
  - Classify with Bayes Rule
    \[
P(C_1 | x) = \frac{P(x | C_1)P(C_1)}{P(x | C_1)P(C_1) + P(x | C_2)P(C_2)}\]
    - Need to estimate the likelihoods and prior terms
Classification: The Bayesian Way (cont’d)

- Determine prior probabilities
  - Collect data at Gate 1: ask drivers how much their cars cost, and measure height of car
  - Result of 1209 samples: \#C1=221 \#C2=988

$$P(C_1) = \frac{221}{1209} = 0.183$$

$$P(C_2) = 1 - P(C_1) = 0.817$$
Classification: The Bayesian Way (cont’d)

- Determine likelihoods (class conditional probabilities)
  - Discretize car height into bins, and use normalized histograms (bins are centers at 0.7m, 0.8m, etc.)
Classification: The Bayesian Way (cont’d)

- Calculate the posterior probability for each bin with Bayes rule:

\[
P(C1 \mid x = 1.0) = \frac{P(x = 1.0 \mid C1)P(C1)}{P(x = 1.0 \mid C1)P(C1) + P(x = 1.0 \mid C2)P(C2)}
\]

\[
= \frac{0.2081 \times 0.183}{0.2081 \times 0.183 + 0.0597 \times 0.817}
\]

\[
= 0.438
\]
Classification: The Bayesian Way (cont’d)

- **Pros of Bayesian classification**
  - Optimal if distributions can be estimated reliably
  - The optimal answer is easy to determine from the class with maximal posterior probability
  - Can be extended to include loss functions, i.e., functions that consider more than just the posterior probability
  - Allows to include prior knowledge in a natural way

- **Cons of Bayesian classification**
  - Probability distributions are hard to estimate in high dimensional spaces with continuous features
  - A lot of effort is spent on estimating distributions, but all that matters is the decision boundary
Discriminant Functions

- The decision boundary (discriminant function) between two classes is defined as boundary in feature space that fulfills:

\[ P(C_1 \mid x) = P(C_2 \mid x) \]

or

\[ P(x \mid C_1)P(C_1) = P(x \mid C_2)P(C_2) \]

- What is the optimal decision boundary?
  - The boundary that minimizes misclassification
Discriminant Functions

- Decision boundaries need not be defined in terms of probabilities, but can just be a general discriminant function:
  
  Choose class $C_i$ if $y_i(x) > y_j(x)$ for all $j \neq i$

- Obviously, the posterior probability can be used as discriminant function

  \[
  y_i(x) = P(C_i | x) = P(x | C_i)P(C_i)
  \]

  or

  \[
  y_i(x) = \ln P(x | C_i) + \ln P(C_i)
  \]

  - Note that the decision boundary is invariant if the discriminant function is transformed by a monotonic function

- But we can also create discriminant function directly:

  \[
  y(x) = f(x,w)
  \]
Learning Discriminant Functions

- Motivation
  - Avoid explicit probability estimations, just try to find the decision boundary directly
  - Learning discriminant functions is a function approximation problem
  - Learning parameters of function from examples

- A natural question:
  - What kind of discriminant function can be realized by a certain function approximator? Decision boundaries can be quite complex …
Learning Linear Discriminant Functions

- Two classes (binary classification)

\[
y(x) = f(x, w) = w^T x + w_0
\]

Can only realize linear decision boundaries
Learning Linear Discriminant Functions

- Decision boundary is given by
  \[ y(x) = w^T x + w_0 = 0 \]

  Thus, a decision boundary is \( d-1 \) dimensional in feature space, e.g.,
  - A point for a 1D feature space
  - A line for a 2D feature space
  - A plane for a 3D feature space
  - A hyperplane for \( d>3 \)

- Example:
Learning Linear Discriminant Functions

- Multiple Classes
  \[ y_i(x) = w_i^T x + w_{i,0} \]

- Decision boundaries
  \[ y_i(x) = y_j(x) \]
  \[ w_i^T x + w_{i,0} = w_j^T x + w_{j,0} \]
  \[ (w_i - w_j)^T x + w_{i,0} - w_{j,0} = 0 \]
Learning Linear Discriminant Functions

- Decision Regions

Can we just build a k-class classifier from combining several binary classifiers?
Learning Linear Discriminant Functions

- Decision Regions must be simply connected (no holes) and convex!!!

Given: \( x^A, x^B \in R_k \)

Any point on a line between \( x^A \) and \( x^B \) fulfills:
\[
\hat{x} = \alpha x^A + (1 - \alpha) x^B \quad \text{for} \quad \alpha \in [0, 1]
\]

Since:
\[
y_k(\hat{x}) = \alpha y_k(x^A) + (1 - \alpha) y_k(x^B)
\]

And:
\[
y_k(x^A) > y_j(x^A) \quad \text{and} \quad y_k(x^B) > y_j(x^B) \quad \text{for all} \quad i \neq j
\]

And the decision boundaries are linear.

If follows that:
\[
y_k(\hat{x}) > y_j(\hat{x})
\]

Thus: \( R_k \) is convex and simply connected.
Least Squares Discrimination

- Why not use least squares regression for multi-class classification?

\[
y_i = w_i^T \tilde{x} + w_{i,0}
\]

\[
y(x) = Wx
\]

\[
t_i = [0,0,\ldots,0,1,0,\ldots,0]^T
\]

\[
W = \left(X^T X\right)^{-1} XT
\]

However, least squares classification is not robust!

\[\Rightarrow\] Logistic regression is better!
Summary

- If we know prior and class conditional distributions, Bayes rule tells us how to do optimal decisions based on the posterior probabilities.
- However, for classification, distributions are not required, just the discriminant functions are needed.
- Linear learning systems can approximate linear discriminant function (and convex discriminant regions) without estimating the prob. distributions.
- Least squares classification, however, is not a good approach \(\Rightarrow\) the least squares cost function is not correct for classification.