Bayes Classifiers II: More Examples

CAP5610 Machine Learning
Instructor: Guo-Jun Qi
Recap: Bayes Classifier

• MAP rule: given an input feature vector $X=(X_1, X_2, ..., X_N)$ with $N$ attributes, the optimal binary prediction on its class $Y$ is made by

$$Y^* = \text{argmax}_{Y \in \{0, 1\}} P(Y|X)$$

• Bayes rule: $P(Y|X) \propto P(X|Y)P(Y)$
  • where $P(X|Y)$ is the class-conditional distribution for class $y$, and
  • $P(Y)$ is the prior distribution.
Bayes error

• Two types of prediction error

\[
p(error|X) = \begin{cases} 
    p(Y = C_1|X), & \text{if } P(Y = C_2|X) > P(Y = C_1|X) \\
    p(Y = C_2|X), & \text{if } P(Y = C_1|X) > P(Y = C_2|X) 
\end{cases} 
= \min\{p(Y = C_1|X), p(Y = C_2|X)\}
\]

• An example with one dimensional X
Bayes error

• The shaded area under the curve corresponds to the prediction errors incurred by the Bayes classifier.
Optimality

• Bayes classifier is the optimal classifier we can obtained with the smallest prediction error.

• The prediction error of NN classifier is upper bounded by twice Bayes error asymptotically (i.e., when the size of training set is very large).
Decision region and boundary

- Log likelihood ratio divides the feature space into two regions by threshold 0.

\[
\log \frac{P(Y=1|X)}{P(Y=2|X)} > 0
\]

\[
\log \frac{P(Y=2|X)}{P(Y=1|X)} < 0
\]
The boundary

• is determined by the equation:

\[
\log \frac{P(Y = 1|X)}{P(Y = 2|X)} = \log P(Y = 1|X) - \log P(Y = 2|X) = 0
\]

• Discriminant function \( f(X) = \log P(Y = 1|X) - \log P(Y = 2|X) \)
Examples of binary decision boundary

• Gaussian class-conditional density (one dimensional $X$)

$$P(X = x|Y = y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp \left( -\frac{(x - \mu_y)^2}{2\sigma_y^2} \right)$$

with $y = \{1,2\}$. 
High dimensional case

• Gaussian class-conditional density in high dimensional space

\[
P(X = x | Y = y) = \frac{1}{\sqrt{2\pi|\Sigma_y|}} \exp \left( -\frac{(x - \mu_y)^T \Sigma_y^{-1} (x - \mu_y)}{2} \right)
\]

• Decision boundary equation:

\[
f(X) = \log \left( \frac{P(Y = 1) P(X|Y = 1)}{P(Y = 2) P(X|Y = 2)} \right)
= -\frac{(X - \mu_1)^T \Sigma_1^{-1} (X - \mu_1)'}{2} - \frac{(X - \mu_2)^T \Sigma_2^{-1} (X - \mu_2)'}{2} + \frac{1}{2} \log \frac{|\Sigma_2| P(Y = 1)}{|\Sigma_1| P(Y = 2)} = 0
\]

• A quadratic surface in high dimensional space
Special Case

• If $\Sigma_1 = \Sigma_2 = \Sigma$, discriminate function boils down to

$$f(X) = -X \sum^{-1} (\mu_1 - \mu_2) + \mu_1 \sum^{-1} \mu_1 - \mu_2 \sum^{-1} \mu_2 + \log \frac{P(Y = 1)}{P(Y = 2)}$$

$$= XW + b$$

where $W$ is a N dimensional vector, $b$ is a real number.

• $f(X)$ is linear! Decision boundary is a hyper plan in high-dimensional space.
Decision boundary

- We got the first linear classifier from Bayes classifier model.
Linear classifier

- $W$ and $b$ is indirectly obtained from two class-conditional distribution

$$f(X) = -X \Sigma^{-1} (\mu_1 - \mu_2) \cdot w + \mu_1 \Sigma^{-1} \mu_1 - \mu_2 \Sigma^{-1} \mu_2 + \log \frac{P(Y = 1)}{P(Y = 2)}$$

$$= XW + b$$

- We waste much effort on estimating Gaussian covariance matrix and mean vector, but what we really need is simply $W$ and $b$, can we learn $W$ and $b$ directly? Yes!

- Directly estimating $W$ and $b$ reduces the number of parameters we have to estimate.
Bayes classifier for continuous feature vectors

• Input feature vector \( X = (X_1, ..., X_N) \) with \( N \) attributes
• Step 1: design class-conditional density for \( Y \in \{0, 1, ..., 9\} \)

\[
P(X = x | Y = y) = \frac{1}{\sqrt{2\pi|\Sigma_y|}} \exp\left(-\frac{(x - \mu_y)\Sigma_y^{-1}(x - \mu_y)'}{2}\right)
\]

• Naïve Gaussian Bayes classifier

\[
P(X|Y) = P(X_1|Y) \ldots P(X_N|Y)
\]

Where each individual term is

\[
P(X_n | Y = y) = \frac{1}{\sqrt{2\pi\sigma_{y,n}}} \exp\left(-\frac{(X_n - \mu_{y,n})^2}{2\sigma_{y,n}^2}\right)
\]
Bayes classifier for continuous feature vectors

• Maximum Likelihood estimation of $\mu_{y,n}$ and $\sigma_{y,n}$ for

$$P(X_n | Y = y) = \frac{1}{\sqrt{2\pi}\sigma_{y,n}} \exp \left( -\frac{(X_n - \mu_{y,n})^2}{2\sigma_{y,n}^2} \right)$$

are

$$\mu_{y,n} = \frac{1}{m_y} \sum_{i=1}^{m} X_{i,n} \delta Y_i = y$$

$$\sigma_{y,n} = \frac{1}{m_y - 1} \sum_{i=1}^{m} \delta Y_i = y \ X_{i,n} - \mu_{y,n}^2$$

with a set of $(X_i, Y_i)$ for ith training example, and $X_{i,n}$ is the nth feature for $X_i$, $m_y$ is the number of training examples of class $y$, and $\delta [Y_i = y]$ is indicator function.
Bayes classifier for continuous feature vectors

• Step 2: Estimate the prior distribution

\[ P(y = d) = \theta_d \text{ with } \sum_{d=0}^{9} \theta_d = 1 \]

• Solution 1: Maximum likelihood estimation

\[ \theta_d = \frac{\# \text{ of digit } d \text{ in training set}}{\# \text{ of total digits in training set}} \]
Bayes classifier for continuous feature vectors

• Solution 2: Maximum A Posterior parameter estimation
  • Imposing a prior $P(\theta)$ on the parameter of prior distribution $P(y|\theta)$: Prior on prior
  • Instead of only estimating a single point for $\theta$, we estimate a posterior distribution $P(\theta|D_Y)$ over all possible $\theta$, where $D_Y$ is the class labels for training examples
  • Dirichlet distribution $P(\theta) \propto \theta_1^{\alpha_1-1} \ldots \theta_9^{\alpha_9-1}$, conjugate to $P(y|\theta)$
  • By the property of conjugate distribution, we have
    $$\arg\max_{\theta_d} P(\theta_d|D_Y) = \frac{\# \text{ of digit } d \text{ in training set} + \alpha_d}{\# \text{ of training examples} + \sum_{d=0}^{9} \alpha_d}$$
Bayes classifier for continuous feature vectors

• Test example $X$, its most possible digit is

$$Y^* = \arg\max_{Y \in \{0, 1, \ldots, 9\}} P(X_1|Y) \cdots P(X_N|Y)P(Y)$$

$$= \arg\max_{Y \in \{0, 1, \ldots, 9\}} \log P(X_1|Y) + \cdots + \log P(X_N|Y) + \log P(Y)$$

• A trick: working with log of probability
  • Convert multiplication to summation
  • Avoid arithmetic underflow: too large $N$ will cause too small posterior that cannot be correctly operated by computer.
Machine Problem 2

• Applying the Naïve Gaussian Bayes classifier on MNIST.
  • Use two solutions (MLE/MAP) to estimate prior distribution $P(Y)$
  • Estimate the independent Gaussian density for each dimension of feature vector
  • Report error rates with three test protocol, you need to tune $\alpha_d$ for $d=0,...,9$
    • Test/training
    • Test/validation/training
    • 5-fold/10-fold cross-validation
    • Tune $\alpha_d = 1\%, 2\%, 4\%, 8\%, 16\%$ of training examples.
  • Compare your result with KNN, which is better? Why?
  • KNN lost correlations between different features (pixels in MNIST case)?
  • Do you observe over fitting? Do $\{\alpha_d\}$ help to reduce over fitting?
Machine Problem 2

• Due in three weeks (11:59am, before Thursday’s class, Sep 18)
• Submit to cap5610ucf@gmail.com.
• Report, and source code
Why not using multivariate Gaussian

• With full covariance matrix to model the class-conditional distribution?

• In MNIST, feature space dimension N=28X28, how many parameters are there in a full covariance matrix?
  • $\frac{N(N+1)}{2} = 307,720$, compared with 50000 training examples
  • Underdetermined: The parameters cannot be completely determined.
Text document: length-varying feature vectors

• Input
  • For each document $m$, a vector of length $n$ represents all the words appearing in the document: $X_n^m$ is the $n$-th word in document $m$.
    • The domain of $X_n^m$ is a vocabulary of a dictionary (e.g., Webster dictionary)
    • The length of documents vary between each other, so feature vector $X^m = (X_1^m, X_2^m, \ldots)$ for a document does not have a fixed size.

• Output
  • $Y_m$ defines the category of document $m$ – {Ads, Non-Ads}
Class-conditional distribution

• Assume independence between words in a document

\[ P(X_1^m, X_2^m, \ldots, X_n^m | Y_m) = P(X_1^m | Y_m) P(X_2^m | Y_m) \ldots P(X_n^m | Y_m) \]

• A graphical model representation for the independence decomposition of a distribution
  • Each circle node represent a random variable
  • Each arrow represents a conditional distribution, conditioned on the parent node
Class-conditional distribution

• Class-conditional distribution

\[
P(X_1^m, X_2^m, ..., X_n^m | Y_m) = P(X_1^m | Y_m)P(X_2^m | Y_m) ... P(X_n^m | Y_m) = \prod_{i=1}^{V} P(w_i | Y_m)^{\text{count}_i}
\]

• MLE of \( P(w_i | Y_m) \) for each word \( w_i \) in a vocabulary.

\[
P(w_i | Y_m = y) = \frac{\# \text{ of word } w_i \text{ in documents of class } y}{\# \text{ of words in documents of class } y}
\]
Class-conditional distribution

• MAP estimate of $P(w_i|Y_m)$ for each word $w_i$ in a vocabulary.

$$P(w_i|Y_m = y) = \frac{\text{# of word } w_i \text{ in documents of class } y \text{ and soft count } \alpha_i}{\text{# of words and all soft counts in documents of class } y}$$

• Here a soft count $\alpha_i$ is added to each word $w_i$
Class prior

• Similar MLE/MAP methods can be applied to estimate $P(Y)$
• Given a test example $X$,

\[
P(X_1, X_2, \ldots, X_n | Y) = \prod_{i=1}^{V} P(w_i | Y)^{\text{count}_i}
\]

• Decide the optimal $y$

\[
\arg\max_{y} P(Y = y) \prod_{i=1}^{V} P(w_i | Y = y)^{\text{count}_i}
\]
Summary

• Recap Bayes classifier, Bayes error
• Decision region and boundary with Gaussian class-conditional distributions
  • Linear hyper plane and quadratic surface in high dimensional space
• Gaussian Naive Bayes, Machine Problem 2
• Bayes classifier with length-varying feature vector for text document
• Basic concept of graphical model