Lecture 6: Classification Methods I

James D. Wilson

MATH 373

Plan for this Lecture

- Bayes Classifiers
- Naïve Bayes Classifiers
- K-Nearest Neighbors (K-NN) Classification

Reference: ISL Sections 2.2.3; 4.4

Recall: Classification Setting

Training Data: Consisting of n observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

- y_i are discrete valued observations

Test Data: Observations of the form $(\mathbf{x}_o, y_o)$.

Goal:

- Train a classifier $\phi(x) = \hat{y}$ using the training data.
- Identify the classifier that minimizes the MSPE on the test data:

$$\text{Ave}(\mathbb{I}(y_o \neq \hat{y}_o))$$

The Bayes Classifier

Theorem

Minimizing $\text{Ave}(\mathbb{I}(y_o \neq \hat{y}_o))$, on average, is equivalent to choosing the class j for which the quantity

$$\mathbb{P}(Y = j \mid X = \mathbf{x}_o)$$

is largest.

The classifier $\phi(\mathbf{x}_o) = \text{argmax}_j \left(\mathbb{P}(Y = j \mid X = \mathbf{x}_o)\right)$ is the Bayes Classifier.

Key Question: How do we calculate the Bayes Classifier, and what exactly do we mean by this *conditional probability*?

Stochastic Setting (Binary case)

Regard observations $(X_1, Y_1), \ldots, (X_n, Y_n)$ as being independent samples from a fixed distribution $\mathbb{P}$ on $\mathcal{X} \times \{-1, +1\}$

Notation: use (X, Y) to denote a generic pair with distribution $\mathbb{P}$ and independent of the observations.

Quantities of Interest (Bayesian statistics revisited...)

1. Prior probabilities of $Y = +1$ and $Y = -1$
2. Conditional probability of $Y = +1$ given $X = \mathbf{x}$
3. Class conditional distributions of X given $Y = y$

Prior Probabilities of Y (Binary case)

Let $\pi_{-1} = \mathbb{P}(Y = -1)$ and $\pi_1 = \mathbb{P}(Y = +1)$

- Probability of seeing class $Y = -1$ or $Y = +1$ *before* (prior to) observing $\mathbf{x}$
- Relative abundance of class -1 and $+1$
- Note $\pi_{-1} + \pi_1 = 1$
- Cases in which $\pi_{-1} >> \pi_1$ or v.v. can be problematic (problem of unbalanced data)

Unconditional and Conditional Densities of X

Assume: $\mathcal{X} \subseteq \mathbb{R}^p$ and X has unconditional joint density $f(\mathbf{x})$:

$$\mathbb{P}(X \in A) = \int_A f(\mathbf{x})\, d\mathbf{x}, \quad A \subseteq \mathcal{X}.$$

Let $f_y(\mathbf{x})$ denote class-conditional density of X given $Y = y$.

$$\mathbb{P}(X \in A \mid Y = y) = \int_A f_y(\mathbf{x})\, d\mathbf{x}, \quad A \subseteq \mathcal{X}.$$

Take-away: Class-conditional densities f_{-1} and f_1 tell us about separability of the classes -1s and $+1$s.

Conditional Distribution of Y Given X (Binary case)

Conditional probability of Y given $X = \mathbf{x}$:

$$\eta(\mathbf{x}) = \mathbb{P}(Y = +1 \mid X = \mathbf{x})$$
$$= \text{probability of seeing class } Y = +1 \text{ after observing } \mathbf{x}$$

Note: $\mathbb{P}(Y = -1 \mid X = \mathbf{x}) = 1 - \eta(\mathbf{x})$.

Regimes:

- $\eta(\mathbf{x}) \approx 1 \;\Rightarrow\; Y$ is likely to be $+1$
- $\eta(\mathbf{x}) \approx 0 \;\Rightarrow\; Y$ is likely to be -1
- $\eta(\mathbf{x}) \approx 1/2 \;\Rightarrow\;$ value of Y uncertain

The Bayes Classifier

For binary classification, the Bayes classifier for new data x_o is:

$$\hat{y}_o = \begin{cases} -1 & \text{if} \quad \eta(x) < 0.5 \\ +1 & \text{if} \quad \eta(x) > 0.5 \end{cases}$$

Mathematical Fact: The Bayes classifier $\hat{y}_o$ (for general multi-class classification) has the smallest possible test error rate. This error is called the Bayes error rate and is given by:

$$1 - \mathbb{E}[\max_j \{\mathbb{P}(Y = j \mid X)\}]$$

This value is analagous to the *irreducible error* in regression.

So, the bayes classifier is the best that we can hope to obtain, but...

Bayes Theorem: relationship among distributions

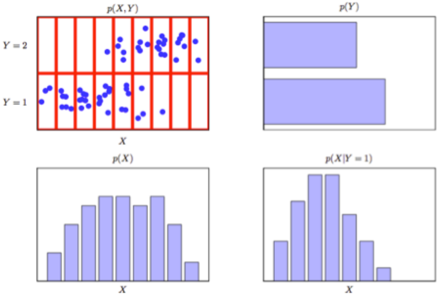
Bayes Theorem gives the following relationship:

$$\mathbb{P}(Y = j \mid X = \mathbf{x}) = \frac{\pi_j f_j(\mathbf{x})}{f(\mathbf{x})} = \frac{\pi_j f_j(\mathbf{x})}{\sum_{j=1}^{m} \pi_j f_j(\mathbf{x})}$$

Key (and unfortunate) point: To obtain the bayes classifier, we need

- Class conditional probabilities: $f_j(\mathbf{x}), j = 1, \ldots, m$
- Prior probabilities $\pi_j, j = 1, \ldots, m$

How do we estimate probabilities?



Two major choices:

- Make assumptions about data. Example: (X, Y) are iid from some distribution
- Empirical estimation of joint density of (X, Y) (i.e. histogram approach)

Summary of Bayes Classifier

- If we knew the class conditional probabilities of X given $Y = y$ and the prior probabilities assoicated with Y, then the Bayes classifier is the best we can do in classification.
- In some applications, it is reasonable to model the above densities based on prior knowledge and statistical inference (e.g., multivariate Guassian for $f_j(\mathbf{x})$)
- In the applications that we cannot provide a model, we have to *estimate* these probabilities.
  - Easily done for π_j:

$$\hat{\pi}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i = j)$$

  - The joint pdf of $f_j(\mathbf{x})$ is challenging without further assumptions...

Naïve Bayes Classifiers

Recall: $f_j(\mathbf{x}) = f(\mathbf{x} \mid Y = j)$

Major (simplifying) Assumption: given Y, features / predictors are conditionally independent of one another:

$$f_j(\mathbf{x}) = \prod_{k=1}^{p} f(x_k \mid Y = j)$$

Result: $f(x_k \mid Y = j)$ can be easily estimated via an empirical (histogram) approach. This is significantly easier than estimating the full joint density of $f_j(\mathbf{x})$

Algorithm

Given: Training observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$, test observation $\mathbf{x}_o$

Estimate:

- $f(x_k \mid Y = j)$ for all **variables** $k = 1, \ldots, p$, and **classes** $j = 1, \ldots, m$
- π_j and $f_j(\mathbf{x}) = \prod_{k=1}^{p} f(x_k \mid Y = j)$ for all j

Calculate:

$$\hat{\mathbb{P}}(Y = j \mid X = \mathbf{x}_o) = \frac{\hat{\pi}_j \hat{f}_j(\mathbf{x}_o)}{\sum_{j=1}^{m} \hat{\pi}_j \hat{f}_j(\mathbf{x}_o)}, \qquad j = 1, \ldots, m$$

Return: Classifier $\phi(\mathbf{x}_o)$ where

$$\phi(\mathbf{x}_o) = \text{argmax}_j(\hat{\mathbb{P}}(Y = j \mid X = \mathbf{x}_o))$$

Event Models for Naïve Bayes

In some cases, it is reasonable to model the class conditional distributions using well-established probabilistic models (think back to your favorite probability course).

For example, consider cases where $X \mid Y = y$ is

- Continuous $\rightarrow$ Gaussian RV
- Count the occurrence of each feature $\rightarrow$ Multinomial RV
- Observation of a feature as a binary variable $\rightarrow$ Bernoulli RV

Example: Gaussian Naïve Bayes

- Assume the likelihood of the features is Gaussian
- Use a parametric likelihood function of real-valued variable X

$$f_i(x) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x-\mu_j)^2}{2\sigma_i^2}}$$

where $\mu_j := \mathbb{E}[X \mid Y = j]$ is the conditional mean and $\sigma_j^2 := \mathrm{Var}(X \mid Y = j)$ is the conditional variance of X given $Y = j$

- The posterior probability is evaluated as a product of univariate conditional density functions

$$\mathbb{P}(Y = j \mid X = \mathbf{x}) \propto \pi_j \prod_{i=1}^{p} f_i(x)$$

Example: Multinomial Naïve Bayes

- X vectors represent the frequencies with which certain events (one per feature) have been generated by a multinomial $(p_1, p_2, \ldots, p_p)$

Example: Probabilities of words appearing in documents

- Documents represented as counts for words that appear in it
- Independence assumption is that the presence of a word is conditionally independent of the presence of another one, given y

Example: Bernoulli Naïve Bayes

Longer name: multivariate Bernoulli

- X vectors are binary variables

Example: $Y = 1$ if a word appears

- Document represented as binary feature vector
- Independence assumption means the presence of a word is conditionally independent of the presence of another one, given Y

Example: Empirical Naïve Bayes

Example: Spam in the Enron Email Corpus

You'd like to develop a spam filter based on the words in the Enron emails from the Enron email directory in 2001. These emails have already been filtered into `spam` emails and `normal` emails. In particular, you'd like to build a filter based on if the word "meeting" is in a new email.

Data is available at `https://www.cs.cmu.edu/~enron/`.

Example: Spam Filter for Individual Words

Digging into the data, you calculate the following empirical probabilities:

- $\hat{\mathbb{P}}(\texttt{spam}) = 0.29$
- $\hat{\mathbb{P}}(\texttt{normal}) = 0.71$
- $\hat{\mathbb{P}}(\text{"meeting"} \mid \texttt{spam}) = 0.0106$
- $\hat{\mathbb{P}}(\text{"meeting"} \mid \texttt{normal}) = 0.0416$

Thus, we can directly obtain:

$$\hat{\mathbb{P}}(\texttt{spam} \mid \text{"meeting"}) = \frac{0.0106 * 0.29}{(0.0106 * 0.29 + 0.0416 * 0.71)} = 0.09 = 9\%$$

Naïve Bayes Classifier Review

- Approximation of the Bayes Classifier
- Assumes that $X_i \mid Y = y, i = 1, \ldots, n$ are independent
- Easy to implement
- Requires a choice of models for the prior distribution of Y and the class-conditional distribtuion of X given $Y = y$.
- Requires thresholding to determine classification

K-Nearest Neighbors

- The Bayes classifier serves as a ”gold standard” of classifiers in that is the best that we can do; yet, the solution is unattainable.
- Naïve Bayes provides an approximation to the Bayes classifier via a simplifying assumption on the class-conditional densities of X given $Y = y$.
- The K-Nearest Neighbors (KNN) classifier also estimates $\mathbb{P}(Y = j \mid X = \mathbf{x})$, but avoids the joint density of X

K-Nearest Neighbors

Algorithm

Given: Positive integer $K \in \{1, \ldots, n\}$, test observation $\mathbf{x}_o$

Identify:

$$\mathcal{N}_o = \{K \text{ points in the training data that are } \textit{closest} \text{ to } \mathbf{x}_o\}$$

Estimate:

$$\hat{\mathbb{P}}(Y = j \mid X = \mathbf{x}_o) = \frac{1}{K} \sum_{i \in \mathcal{N}_o} \mathbb{I}(y_i = j)$$

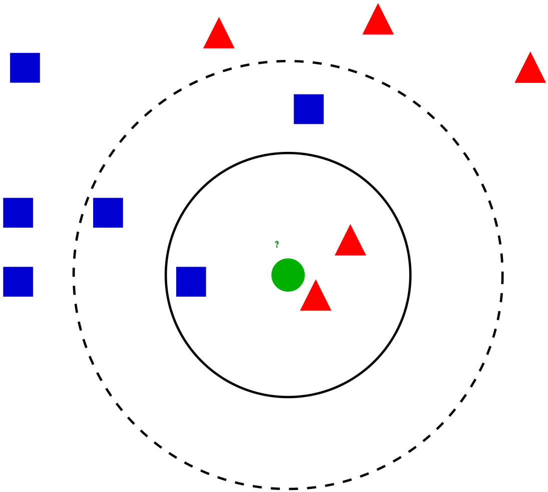
Return: Classifier $\phi(\mathbf{x}_o)$ where

$$\phi(\mathbf{x}_o) = \text{argmax}_j(\hat{\mathbb{P}}(Y = j \mid X = \mathbf{x}_o))$$

K-Nearest Neighbors

- Instance-based learning method: store, or “memorize” training observations
- To classify new data, search memory for observations *near* it
- Pick the majority class (vote) of those observations
- K is a tuning parameter for the algorithm: the number of "neighbors" to search
- Dates back to at least 1951 (Fix and Hodges)

Example



Important Considerations

- How do we know what points are "close"?
- Which K do we use?
- Scaling and Normalization
- Weighted K-NN?

We would like to use a distance measure $d(\cdot, \cdot)$ which satisfy:

1. (**Non-negativity**): $d(x, y) \geq 0$
2. (**Identity**): $d(x, y) = 0$ if and only if $x = y$
3. (**Symmetry**): $d(x, y) = d(y, x)$
4. (**Triangle Inequality**): $d(x, z) \leq d(x, y) + d(y, z)$

Example: Euclidean distance

Most commonly used distance is the Euclidean distance:

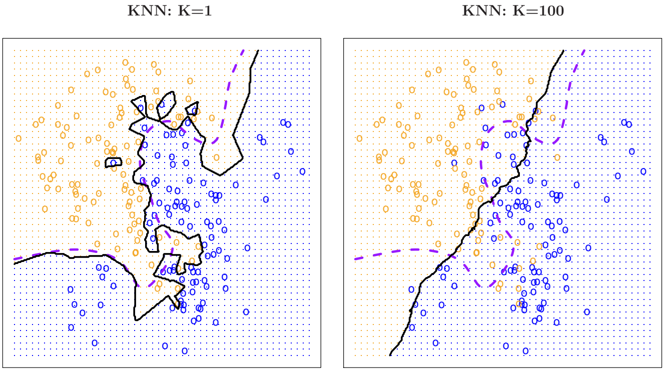
$$\begin{aligned} d(\mathbf{x}_j, \mathbf{x}_k) &= \sqrt{\sum_{i=1}^{p} (x_{j,i} - x_{k,i})^2} \\ &= \sqrt{(\mathbf{x}_j - \mathbf{x}_k)^\top (\mathbf{x}_j - \mathbf{x}_k)} \end{aligned}$$

This is well-suited for continuous data. How about categorical features?

Other Distances

- Pearson correlation - correlation coefficient
- Manhattan distance - distance between two points on a grid
- Levenshtein distance - measure of 'edit distance' used between words (NLP)
- Hamming distance - measure the edit distance between words of the same length
- many more... see https://en.wikipedia.org/wiki/Distance

Effect of K



- Larger K gives smoother boundary
- When K is too large, we always predict the majority class

Effect of K (Extreme example)

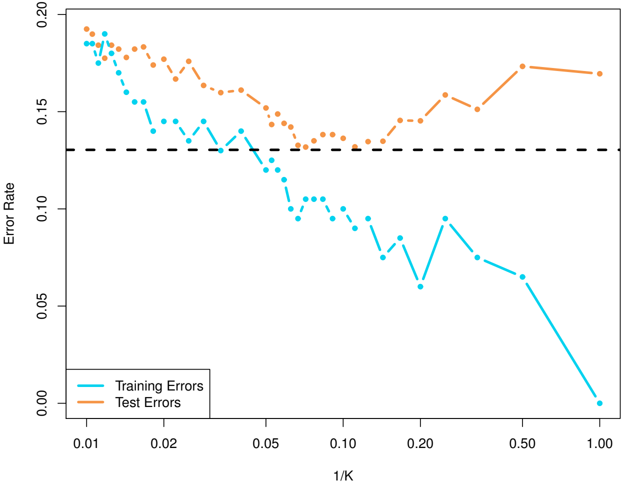


Figure: Black dashed line is Bayes error rate

Accuracy of K-Nearest Neighbor Classifiers

Choice of K affects the accuracy:

- If K is too small, then the result can be sensitive to noise points (overfitting issue).
- If K is too large, then the neighborhood may include too many points from other classes and blur the boundaries (majority voting issue)
- In binary (two class) classification problems, choose K to be an odd number as this avoids tied votes

Accuracy of K-Nearest Neighbor Classifiers

Another issue: Majority voting –

- Can be a problem if the nearest neighbors vary widely in their distance and the closer neighbors more reliably indicate the class of the example.
- One solution is to weight each examples's vote by its similarity, so the vote of an example $\mathbf{x}_i$ for its class is

$$1 - d(\mathbf{x}_i, \mathbf{x}_{new})$$

Weighted majority voting decreases sensitivity of classifier to K

Scaling

- Potential issue when features have different scales
- One feature can dominate distance
- Not always obvious which scaling approach we should use
  - What effect will scaling have on your chosen distance?
  - Standard normalization is regularly used with the Euclidean distance.

K-NN for regression

We can also use K-NN for regression predictions! (and you thought we were done with regression)

Idea: Given test observation $\mathbf{x}_o$ and training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

Identify:

$$\mathcal{N}_o = \{K \text{ points in the training data that are } \textit{closest} \text{ to } \mathbf{x}_o\}$$

Estimate:

$$\hat{y} = \frac{1}{K} \sum_{i \in \mathcal{N}_o} y_i$$

Strengths of K-NN

- K-NN is an easy to understand and easy to implement supervised learning technique
- K-NN is particularly well suited for multi-modal classes
- Often successful when the decision boundary is very irregular
- Training is easy & fast!

Weaknesses of K-NN

- Choice of K
- Model needs to be "re-trained" for each test observation
- Scaling issue
- Accuracy is sensitive to K and to 'majority voting'