MATH 748: Weekly Report

Oct 11-17, 2021

Nayeong Kim (nkim10@sfsu.edu)

8 KNN

8.2 KNN Classifier

(1) Idea of KNN Classifier
The idea of KNN(k Nearest Neighbors) is to classify object based on k closest training examples in the sample space.

(2) Drawback of Majority Vote

- Problem of imbalanced data: More likely to predict to dominant classes.
- One way to overcome: Weighted vote by considering the distance.

(3) Formal Estimator

$$\tilde{Y}(X) = \frac{1}{k} \sum_{x_j \in N(x)} y_j = \text{avg}\{y_j : x_j \in N_k(x)\}$$

$$\hat{f}(X) = \begin{cases} 1 & \tilde{Y}(X) > 0.5 \\ 0 & \tilde{Y}(X) < 0.5 \end{cases}$$

If we set $k = |N(x)|$ to be even, tie can be happened. For binary classifier, k is better to be odd to avoid the situation.

(4) Distance
In this sense, the distance of two points (in the given dataset) is a metric of measuring the similarity of the two points. Various metrics can be used for various purposes. For example, in text cassification, overlap metric, Hamming distance, or cosine distance are used. The choice of metrics can make different results.

- Euclidean distance:
  The most common metric is Euclidean distance. $d(X,X') = \sqrt{\sum (x_j - x'_j)^2} = \sqrt{(X-X')^T (X-X')}$
- Hamming distance:
  The Hamming distance between two equal-length strings of symbols is the number of positions at which the corresponding symbols are different. [1]
- Cosine distance:
  $\frac{X^T X'}{\|X\| \|X'\|} = \cos\theta$

It is recommended to standardize the data with the mean and standard deviation to avoid the case that some data has a larger scale than others. The performance of KNN algorithm can be sigmificantly improved if the distance metric is learned with specified algorithms such as *Large Margin Nearest Neighbor*.

(5) Advantages

- Non-parametric. In other words, it doesn't rely on string-out assumptions about the data.
- It has a low bias in general and has very flexible boundary.
- Robust against outlier.

(6) Limitations

- Not smooth for small k. (Small k has larger variation.)
- Potentially high variance (especially for small k).
- Works well for large n, small p. But not good for small n, large p (high-dimension case). Think of *the curse of dimensionality.*
- Accuracy of KNN can be severely degraded by the presence of noisy or irrelevant features.

8.3 Comparison: Linear Models vs KNN for Binary Classifier

(1) KNN is an approximation to Bayes classifier.

(2) Role of k

- For KNN, the effective number of parameters(the number of degrees of freedom) is $\frac{n}{k}$.
- k controls the complexity.
  k is small $\Rightarrow$ low bias but high variance
  k is large $\Rightarrow$ high bias but low variance
- Best k depends on the data. We can use cross-validation to set the hyper-parameter k.

9 Multi-Class Classification

9.1 Set-up

We can set class labels $r \in \{1,2,\ldots,k\}$ and the classifier $f : \mathbb{R}^p \to \{1,2,\ldots,k\}$ and the loss function (which can be defined as a $k \times k$ matrix). The loss function $C(k, j)$ is the cost of classifying a sample in class k to class j. Generally the diagonal elements are set to be zero (because it is the right classification).

9.2 Notations

(1) Prior: $\pi_k = P(Y = k)$
(2) Posterior Distribution: $P(Y = k|X = x) = \frac{P(X=x|Y=k)P(Y=k)}{P(X=x)} = \frac{f_k(x)\pi_k}{g(x)}$
(3) Marginal of X: $g(x) = \sum \pi_k f_k(x)$
(4) Conditional Density of X: $f_k(x) = P(X = x|Y = k)$

9.3 Bayes Classifier

(1) Risk function:
$E_{X,Y}(C(Y, f(X))$. This is the target we want to minimize.

(2) Bayes classifier:
$\phi_\beta(x) = \underset{f}{\text{argmin}}\ E_{X,Y}C(Y, f(X)) = \underset{f}{\text{argmin}}\ E_X E_{Y|X}C(Y, f(X))$

(3) Inner equal cost:
$\phi_\beta(x) = \underset{k}{\text{argmax}}\ P(Y = k|X = x)$. It assign x to the most probable class using $P(Y = k|X = x)$.

(4) Under unequal cost:
$\phi_\beta(x) = \underset{j}{\text{argmax}}\ \sum_{i=1}^{k} C(i, j)P(Y = i|X = x)$. This is more general loss function than (3).

9.4 Discriminating Function

(1) Each $\delta_k(x)$ is associated with each class. $\delta_k(x) \propto P(Y = k|X = x)$. Hence it is enough to compare $\delta_k(x)$.

(2) Decision rule:
Given x, pick k^* such that $\delta_{k^*}(x)$ is the largest.

(3) Decision boundary:
The boundary for class k and class l is $\{x|\delta_k(x) = \delta_l(x)\}$. Globally, the decision boundary is
$\{x|\ \underset{max}{k}\ \delta_k(x) = \delta_j(x) = \delta_i(x)$ for some $i \neq j\}$

9.5 Methods for Multi-Class Classification

(1) Main Idea: Divide and Conquer

- Divide: Multiple binary classification.
- Conquer: Making final decision using majority vote principal.

(2) One vs Rest(OVR) / One vs All(OVA)
The approach is considering all the cases of j vs not j.

(3) One vs One(OVO)
The approach is comparing classes one-to-one. Step 1: Solve ${}_kC_2$ different one-vs-one binary classifier(all the cases of i vs j for $\forall i, j \in \{1,2,\dots k\}$. Step 2: Predicts to the class that win most out of the battles.

9.6 Linear Classifier for Multi-Class Problems

(1) Linear Regression: Multiple Class Version

$$Z_{(n\times k)} = X_{(n\times p+1)}\beta_{(p+1\times k)} + \varepsilon_{(n\times k)} \text{ where } \beta = [\beta_1 \dots \beta_k] \tag{1}$$

$$\text{In other words, } Z_k = X\beta_k + \varepsilon_k \tag{2}$$

$$\hat{\beta} = \underset{\beta}{\text{argmin}}\ \text{RSS}(\beta) \tag{3}$$

$$= \sum_{k=1}^{K}\sum_{n=1}^{N}(Z_{ik} - \beta_{0k} - \sum_{j=1}^{p}\beta_{jk}x_{ij})^2 = tr[(Z - X\beta)^T(Z - X\beta)] \tag{4}$$

$$\hat{\beta} = (X^TX)^{-1}X^TZ \tag{5}$$

$$\hat{Z} = X\hat{\beta} = [\delta_1(x)\dots\delta_k(x)] \tag{6}$$

We can choose j such that $\delta_j(x)$ is the largest i.e. $\hat{f}(x) = \underset{j}{\text{argmax}}\ \delta_j(x)$. The limitation of this method is that linear structue may not be valid for some datasets and this is not applicable when k is large and p is small.

(2) LDA
Key assumption for LDA is $X|Y = j \sim N(\mu_j, \Sigma)\ \ j = 1,2,\dots k$. Bayes classifier chooses the k such that $\pi_k f_k(x)$ is the largest. It is the same when we take log. We can interpret LDA using *Mahalanobis Distance* [2]: $d(X,X') = \sqrt{(X-X')^T\Sigma^{-1}(X-X')}$. (Actually Euclidean distance is the special case $\Sigma = I$)
We will discuss the approach that $X|Y = k \sim N(\mu_j, \Sigma_j)$(QDA) later.

(3) Multiple Logistic Model
This approach is a kind of OVR. Define, for $j = 1,\dots,\hat{l},\dots k, k-1$ equations $\log\frac{P(Y=j|X=x)}{P(Y=l|X=x)} = \beta_0^{(j)} + \beta_1^{(j)T}X$.
Then we can get $P(Y=j|X=x) = \frac{e^{h_j(x)}}{1+\sum e^{h_i(x)}}$. Also we can set $P(Y=l|X=x)$ by substracting all the others from 1.

(4) QDA
It is the extention from LDA. Recalling LDS, $X|Y=k$ is assumed to follow $N(\mu_k, \Sigma)$. The variations are assumed to be the same: Σ. But, in QDA, we assume that $X|Y=k \sim N(\mu_k, \Sigma_k)$. While LDA has a linear boundary, QDA has a curvy one. Hence,, LDA sometimes has a better performance than QDA. In LDA, Σ has $O(p^2)$ parameters but, in QDA, Σ_k has $O(kp^2)$ (much more than LDA) parameters.

References

[1] Wikipedia: Hamming Distance

[2] Wikipedia: Mahalanobis Distance