Machine Learning and Data Mining I: Lecture 6
k-Nearest Neighbor Learning and the Gaussian Distribution

Morgan McCarty

12 July 2023

1 ML Recipe Review

1. Select Data →
2. Explore →
3. Transform →
4. Train/Test Split →
5. Build Model →
6. Train Model

2 Factors for Choosing KNN

Overtime data in the domain may shift (domain drift). This can be caused by many factors. KNN is a non-parametric model, meaning it does not make any assumptions about the data. This makes it robust to domain drift.

3 k-Nearest Neighbors

- Uses proximity to make calculations about the groupings of data.
- Assumes that similar data points are close together.

3.1 Simiplified Algorithm

- Given a instance with known features, but unknown label.
- Find the k nearest neighbors to the instance and take a majority vote.
- The majority vote is the predicted label.
- High similarity (low distance) → high probability of being in the same class
- Pick k closest neighbors (k is a hyperparameter) and make a prediction based on the majority class

3.2 Algorithm

1. Given a data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^d$ and $y_i \in V$ where V is a set of discrete labels.
2. Given an instance $x \in \mathbb{R}^d$.
3. Compute the distance between x and each x_i in D.
4. Sort the distances in ascending order.
5. Select the k closest neighbors.
6. Return the majority class of the k closest neighbors.

To note k should be odd to avoid ties.

3.3 Distance

3.3.1 Euclidean Distance

$$d(x, x_i) = \sqrt{\sum_{j=1}^{d} (x_j - x_{ij})^2} \tag{1}$$

- This is the most common distance function. It is the straight line distance between two points.
- Some attributes (e.g. categorical attributes) have to be converted to numerical values.
- Can have labels be continuous, discrete, or categorical.

3.3.2 Manhattan Distance

$$d(x, x_i) = \sum_{j=1}^{d} |x_j - x_{ij}| \tag{2}$$

- This is the distance between two points if you can only travel along the axes.
- This is useful for when you have a lot of dimensions.

3.3.3 Cosine Similarity

$$d(x, x_i) = \frac{x \cdot x_i}{||x|| \cdot ||x_i||} \tag{3}$$

- This is the angle between two vectors.
- This is useful for when you have a lot of dimensions.

3.3.4 Hamming Distance

$$d(x, x_i) = \sum_{j=1}^{d} \delta(x_j, x_{ij}) \tag{4}$$

- This is the number of attributes that are different between two points.
- This is useful for when you have a lot of dimensions.

3.4 Multiple Classes

- If there are multiple classes, the majority vote is the class with the most neighbors.
- If there is a tie (by distance or by count), the class is chosen randomly.
- Again k should be odd to avoid ties.

3.5 Overfitting

- If k is too small, the model will overfit.
- If k is too large, the model will underfit.
- Higher k values remove smaller subregions which can lead to underfitting.

3.6 Choosing k

- k is a hyperparameter.
- k is usually chosen by cross-validation.
- Plot the error rate vs k and choose the k with approximately the lowest error rate.

3.7 Intelligibility

- With KNN, it is very easy to show why a decision was made.

3.8 KD-Trees

- It is possible to create a tree structure to store the data.
- This allows very easy and exact determination of why a decision was made.

3.9 Heterogenous/Categorical Attributes

- For heterogenous attributes, we can "one-hot encode" the attributes. A form of normalization.
- E.g. for "Male" v. "Female":

"Male"	"Female"
0	1

  This data point would be "Female".
- For a 3 attribute example of Residential Status ["Owner", "Renter", "Other"]:

"Owner"	"Renter"	"Other"
1	1	0

  This data point would be "Owner", "Renter".
- If it is necessary to have only one true value, you can use a "bit vector".

3.10 Curse of Dimensionality

- Since all features contribute to the distance, the more features there are, the less meaningful the distance is.
- As dimensionality increases the performance of KNN will increase until an optimal point and then decrease towards infinity.

3.11 Weight of Dimensions

- Some dimensions may be more important than others.
- We can weight the dimensions to make them more important.
- This can be done by multiplying the distance by a weight.
- This can be done by adding a weight to the distance.
- This can be represented as:

$$d = w_1|\delta A_i|^r + w_2|\delta B_i|^r + \cdots + w_n|\delta Z_i|^r \tag{5}$$

- Where:
  - w_i is the weight of the ith dimension.
  - δA_i is the difference with respect to feature (dimension) i.
  - r is an exponent

4 The Gaussian Distribution

4.1 Univariate

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{6}$$

- μ is the mean.
- σ^2 is the variance.
- σ is the standard deviation.

4.1.1 Standard Normal

$$\mathcal{N}(x|0, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{7}$$

- $\mu = 0$
- $\sigma^2 = 1$
- $\sigma = 1$

4.1.2 Other Values

- Precision is the inverse of variance. $\beta = \frac{1}{\sigma^2}$
- Log Normal Form (for numerical stability):

$$\ln P(x|\mu, \sigma^2) = \frac{1}{2\sigma^2}\left(\Sigma_{n=1}^{N} x_n - \mu^2 - \frac{N}{2}\ln 2\pi - \frac{N}{2}\ln \sigma^2\right) \tag{8}$$

4.2 Multivariate

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{9}$$

- μ is the mean.
- Σ is the covariance matrix.
- d is the number of dimensions.

4.2.1 Covariance Matrix

- Symmetric matrix.
- Diagonal is the variance of each dimension.
- E.g.

$$P(x_1|x_1) = \begin{bmatrix} \sigma x^2 x & \sigma_{xy} \\ \sigma_{xy} & \sigma y^2 y \end{bmatrix} \tag{10}$$

4.2.2 Mean

$E[X] = \int_x xP(x; \mu, \Sigma)dx = \mu$

4.2.3 Covariance

$E[(X-\mu)(X-\mu)^T] = \Sigma$ (outer product)