KNN

Outline

K-Nearest Neighbors (KNN)

Python Implementation

Math Example

K-Nearest Neighbors (KNN)

What is KNN?

- Non-parametric, instance-based (lazy) learning algorithm
- Used for both classification and regression
- No training phase — stores all training data
- Prediction based on the majority vote (classification) or average (regression) of the *k* nearest neighbors

Algorithm Steps

1. Choose value of k
2. Compute distance between test point and all training points
3. Select k closest neighbors
4. **Classification**: majority vote
   Regression: average of neighbor values

Distance Metrics

- **Euclidean Distance:** $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Manhattan Distance: $\sum_{i=1}^{n} |x_i - y_i|$
- Minkowski Distance (generalized): $\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$

Choosing k and Considerations

- Small k: sensitive to noise (overfitting)
- Large k: more robust but may underfit
- Use cross-validation to choose optimal k
- Normalize features (scaling is critical!)

Pros and Cons

Pros:

- Simple and intuitive
- No training time
- Adapts well to changing data

Cons:

- Slow at prediction time
- Sensitive to irrelevant or correlated features
- Suffers from the curse of dimensionality

Python Implementation

Python Code Example

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X, y = load_iris(return_X_y=True)
X = StandardScaler().fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y)

model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train, y_train)
print("Accuracy:", model.score(X_test, y_test))
```

Summary

- KNN is powerful for low-dimensional, small datasets
- Performance highly dependent on distance metric and feature scaling
- Use cross-validation to tune k
- Consider dimensionality reduction techniques for high-dimensional data

Math Example

KNN Classification: Math Example

Dataset:

Point	x	y	Label
A	1	2	Red
B	2	3	Red
C	3	3	Blue
D	6	5	Blue

Query Point: $Q = (3, 4)$

Choose $k = 3$

Step 1: Compute Euclidean Distances

Euclidean distance: $d(p, q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

- $d(Q, A) = \sqrt{(3-1)^2 + (4-2)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$
- $d(Q, B) = \sqrt{(3-2)^2 + (4-3)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$
- $d(Q, C) = \sqrt{(3-3)^2 + (4-3)^2} = \sqrt{1} = 1.0$
- $d(Q, D) = \sqrt{(3-6)^2 + (4-5)^2} = \sqrt{9+1} = \sqrt{10} \approx 3.16$

Step 2: Nearest Neighbors and Voting

3 Nearest Neighbors (Sorted):

Point	Distance	Label
C	1.0	Blue
B	1.41	Red
A	2.83	Red

Majority Vote:

- Red: 2 votes (B, A)
- Blue: 1 vote (C)

Predicted Class: **Red**

Thank you!