Tutorial: K-Nearest Neighbors

Cheat Sheet and Practice Problems

ES335 - Machine Learning
IIT Gandhinagar

July 23, 2025

1 Summary from Slides

1.1 Basic Concept

K-Nearest Neighbors (KNN): Non-parametric, instance-based learning algorithm

- Stores all training instances
- Makes predictions based on k nearest neighbors
- No explicit training phase - "lazy learning"
- Works for both classification and regression

1.2 Algorithm

For Classification:

1. Calculate distance from query point to all training points
2. Find k nearest neighbors
3. Use majority vote among k neighbors
4. Predict the most frequent class

For Regression:

1. Calculate distance from query point to all training points
2. Find k nearest neighbors
3. Average the target values of k neighbors
4. Predict the average value

1.3 Distance Metrics

Euclidean Distance:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattan Distance:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

Minkowski Distance:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

1.4 Key Properties

Advantages:

- Simple to understand and implement
- No assumptions about data distribution
- Can model complex decision boundaries
- Works well with small datasets

Disadvantages:

- Computationally expensive for large datasets
- Sensitive to irrelevant features
- Sensitive to scale of features
- Performance degrades in high dimensions

1.5 Choosing K

Small K:

- More flexible, can model complex patterns
- Higher variance, sensitive to noise
- Risk of overfitting

Large K:

- Smoother decision boundaries
- Lower variance, more stable
- Risk of underfitting

Rule of thumb: Use odd values to avoid ties, often $k = \sqrt{n}$

2 Practice Problems

Problem : Basic KNN Classification

Given training data:

X1	X2	Class
1	2	A
2	3	A
3	1	B
4	2	B
2	1	A

For query point (2, 2), predict the class using K=3 with Euclidean distance.
a) Calculate distances to all training points b) Identify 3 nearest neighbors c) Make prediction using majority vote

Problem : Distance Metrics Comparison

For points A(1,1) and B(4,5):
a) Calculate Euclidean distance b) Calculate Manhattan distance c) Calculate Minkowski distance with p=3 d) Which distance metric would be most appropriate for different scenarios?

Problem : Effect of K Value

Dataset with 9 points: - 5 points of class A: (1,1), (1,2), (2,1), (2,2), (1.5,1.5) - 4 points of class B: (4,4), (4,5), (5,4), (5,5)
For query point (3,3), predict class using: a) K=1 b) K=3 c) K=5 d) K=7
Explain how predictions change with different K values.

Problem : KNN Regression

Training data for house prices:

Size (sq ft)	Bedrooms	Price ($1000s)
1000	2	150
1200	3	180
1500	3	220
1800	4	280
2000	4	320

Predict price for a house with 1400 sq ft and 3 bedrooms using K=3.
a) Normalize features before calculating distances b) Find 3 nearest neighbors c) Calculate predicted price

Problem : Feature Scaling Impact

Two features with different scales: - Feature 1 (Age): ranges from 20-60 - Feature 2 (Income): ranges from 20000-80000
Query point: Age=30, Income=40000 Training points: (25, 30000, Class A), (35, 45000, Class B)
a) Calculate Euclidean distances without scaling b) Calculate Euclidean distances with standardization c) How does scaling affect the nearest neighbor? d) Why is feature scaling important for KNN?

Problem : Weighted KNN

Implement weighted KNN where nearer neighbors have higher influence:
Weight function: $w_i = \frac{1}{d_i^2}$ where d_i is distance to neighbor i
For the data in Problem 1, calculate weighted prediction for query point (2,2) using K=3.
a) Calculate weights for each of the 3 nearest neighbors b) Compute weighted vote for classification c) Compare with unweighted KNN result

Problem : Cross-Validation for K Selection

You have 20 training samples. Design a cross-validation procedure to select optimal K:
a) What values of K should you test? b) How would you set up 5-fold cross-validation? c) What metric would you use to evaluate different K values? d) How do you handle the case where K ¿ fold size?

Problem : Curse of Dimensionality

Consider KNN performance as dimensionality increases:
Given 1000 points uniformly distributed in a unit hypercube:
a) In 2D: What's the expected distance to nearest neighbor? b) In 10D: How does this distance change? c) Why does KNN performance degrade in high dimensions? d) What preprocessing steps can help mitigate this?

Problem : Computational Complexity

Analyze KNN computational complexity:
For n training samples, d dimensions, and k neighbors:
a) Training time complexity b) Prediction time complexity (naive approach) c) Space complexity d) How can you speed up nearest neighbor search? e) Compare with other algorithms like decision trees

Problem : Handling Categorical Variables

Dataset with mixed features: - Age: 25, 30, 35, 40 - City: NYC, LA, Chicago, Boston - Salary: 50k, 60k, 70k, 80k
a) How do you calculate distance when features are categorical? b) Implement Hamming distance for categorical features c) How do you combine continuous and categorical distances? d) What are alternative approaches for mixed data types?

Problem : Imbalanced Datasets

Dataset with imbalanced classes: - Class A: 900 samples - Class B: 100 samples
Query point has 5 nearest neighbors: 4 from class A, 1 from class B.
a) What does standard KNN predict? b) How can you modify KNN to handle imbalance? c) Implement class-weighted KNN d) What other techniques can address this issue?

Problem : Missing Values

Training data with missing values:

Feature 1	Feature 2	Class
1	2	A
?	3	A
3	?	B
4	2	B

a) How do you calculate distance when values are missing? b) Implement distance calculation ignoring missing dimensions c) What are alternative approaches for handling missing data? d) When would you impute vs ignore missing values?

Problem : Local vs Global Methods

Compare KNN with global methods like logistic regression:
a) When would KNN outperform global methods? b) What type of decision boundaries can KNN model? c) How does sample size affect KNN vs global methods? d) In what scenarios would you choose KNN over other algorithms?

Problem : KNN Variants

Explore different KNN variants:
a) **Radius-based neighbors**: Instead of K nearest, use all neighbors within radius r. How do you choose r? b) **Locally weighted regression**: Use distance-weighted averaging for regression. Implement this approach. c) **Condensed KNN**: Remove redundant training points. Design an algorithm to identify which points to keep. d) Compare these variants with standard KNN.

Problem : Real-world Application

Design a KNN system for movie recommendation:
Users rate movies on 1-5 scale. Predict rating for user-movie pairs.
a) How do you represent users and movies as feature vectors? b) What distance metric is appropriate for this domain? c) How do you handle the cold start problem (new users/movies)? d) What are the scalability challenges with millions of users? e) How would you incorporate temporal information?

Problem : Advanced Challenge

Implement an efficient KNN using KD-trees:
a) Explain how KD-trees work for nearest neighbor search b) What is the time complexity for building and querying KD-trees? c) In what dimensions do KD-trees become ineffective? d) What are alternative data structures (LSH, Ball trees)? e) Implement approximate KNN - when is this acceptable? Design a complete solution comparing exact vs approximate KNN on different datasets.