In the name of GOD.



Sharif University of Technology

Machine Learning

Spring 2025

Hamid R. Rabiee, Zahra Dehghanian

Homework 5 | Deadline: 1404/02/28

1. (15)

   Consider the K-Nearest Neighbors (KNN) algorithm and answer the following parts:

(A) – Distance Metrics

Suppose we have two points $A = (a_1, a_2)$ and $B = (b_1, b_2)$ in $\mathbb{R}^2$ (with $a_1 < b_1$, $a_2 < b_2$, so they are not collinear). Derive the locus of points (x, y) that are equidistant to A and B under each metric below:

(i) **Euclidean (L2) distance:**

$$d((x, y), A) = d((x, y), B)$$

Simplify to find the equation of the boundary line.

(ii) **Manhattan (L1) distance:**

$$|x - a_1| + |y - a_2| = |x - b_1| + |y - b_2|$$

Describe the shape of this boundary (it will be piecewise-linear rather than a single straight line).

Explain qualitatively how the choice of L2 vs L1 changes the decision boundary of a 1-NN classifier in this simple case.

(B) Bias-Variance Tradeoff

Explain how the choice of k (the number of neighbors) affects the bias and variance of a KNN model (for either classification or regression). In particular, discuss what happens in the limits $k = 1$ and $k = n$ (where n is the training set size). Relate your answer to the bias–variance decomposition of prediction error. You may also mention asymptotic consistency results for KNN (e.g., how KNN error approaches the Bayes error for large n with appropriate k).

(C) Kernel Weighting

In KNN regression, one can assign weights to neighbors based on distance. Let the weight of neighbor i be

$$w_i = K(d(x, x_i))$$

where $d(\cdot, \cdot)$ is the chosen distance.

(a) Write the formula for the kernel-weighted KNN regression prediction $\hat{f}(x)$ in terms of w_i and y_i.

(b) Now consider using a Gaussian kernel

$$K(x_i, x) = \exp\left(-\frac{\|x_i - x\|^2}{\sigma^2}\right).$$

Describe qualitatively what happens as $\sigma \to 0$ and as $\sigma \to \infty$.

Solution:

(A) Effect of Distance Metric (Locus of Equidistant Points)

(i) **Euclidean (L2) metric:** We set

$$(x - a_1)^2 + (y - a_2)^2 = (x - b_1)^2 + (y - b_2)^2.$$

Expanding and simplifying:

$$x^2 - 2a_1x + a_1^2 + y^2 - 2a_2y + a_2^2 = x^2 - 2b_1x + b_1^2 + y^2 - 2b_2y + b_2^2,$$

Canceling x^2 and y^2, we get

$$2(b_1 - a_1)x + 2(b_2 - a_2)y + (a_1^2 + a_2^2 - b_1^2 - b_2^2) = 0.$$

This is the equation of a straight line (the perpendicular bisector of segment AB).

(ii) **Manhattan (L1) metric:** The condition is

$$|x - a_1| + |y - a_2| = |x - b_1| + |y - b_2|.$$

Expanding the absolute values results in a piecewise-linear boundary. Geometrically, under L1, circles become diamonds, and the boundary is composed of segments at 45° angles (slopes ± 1).

Qualitative Explanation: With L2 (Euclidean), the 1-NN decision boundary is a straight line. With L1 (Manhattan), the boundary is piecewise-linear and diamond-shaped, altering the geometry of neighborhoods significantly (source).

(B) Effect of k (Bias–Variance Tradeoff)

General Tradeoff: Increasing k increases bias and decreases variance. Decreasing k does the opposite.

Limits:

- $k = 1$: Very low bias, very high variance. Perfectly fits training data, prone to overfitting.
- $k = n$: Very high bias, very low variance. Predicts the same value everywhere (global majority/mean).

Bias–Variance Decomposition: Prediction error = irreducible noise + bias^2 + variance.

Asymptotic Consistency: If $k \to \infty$ but $k/n \to 0$ as $n \to \infty$, KNN becomes consistent and approaches the Bayes optimal error (source) (source).

(C) Kernel-Weighted KNN

(a) The kernel-weighted KNN regression formula is

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} K(d(x, x_i)) y_i}{\sum_{i=1}^{n} K(d(x, x_i))}.$$

(b) With Gaussian kernel:

$$K(x_i, x) = \exp\left(-\frac{\|x_i - x\|^2}{\sigma^2}\right).$$

As $\sigma \to 0$, only the nearest neighbor matters (approaching 1-NN). As $\sigma \to \infty$, all neighbors are weighted equally (approaching $k = n$) (source).

2. (25) Prove that in Principal Component Analysis (PCA), the reconstruction error after projecting onto the first k principal components is equal to the sum of the discarded eigenvalues, where the reconstruction error is computed using the Frobenius norm. Specifically, show:

$$\frac{1}{n}\|X - \hat{X}\|_F^2 = \sum_{i=k+1}^{d} \lambda_i$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ are the eigenvalues of the data covariance matrix and the data matrix X is zero-centered (i.e., each column has mean zero).

Solution:

The Frobenius norm of a matrix A is defined as $\|A\|_F = \sqrt{\text{tr}(A^\top A)}$, and therefore $\|A\|_F^2 = \text{tr}(A^\top A)$.

Given a zero-centered data matrix $X \in \mathbb{R}^{n \times d}$, define the covariance matrix and its corresponding eigen-decomposition:

$$\Sigma = \frac{1}{n} X^\top X = U \Lambda U^\top,$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$, and $U = [u_1, u_2, \ldots, u_d]$ is the orthonormal matrix of eigenvectors satisfying $U^\top U = I$.

Let $U_k = [u_1, u_2, \ldots, u_k] \in \mathbb{R}^{d \times k}$ denote the matrix formed by the first k eigenvectors.

The reconstruction error is:

$$E = \frac{1}{n}\|X - \hat{X}\|_F^2 = \frac{1}{n}\text{tr}((X - \hat{X})^\top (X - \hat{X})).$$

Projection onto the first k components gives:

$$\hat{X} = X U_k U_k^\top \quad \Rightarrow \quad X - \hat{X} = X(I_d - U_k U_k^\top).$$

Substituting:

$$E = \frac{1}{n}\text{tr}\left[(X(I_d - U_k U_k^\top))^\top (X(I_d - U_k U_k^\top))\right] = \frac{1}{n}\text{tr}\left[(I_d - U_k U_k^\top) X^\top X (I_d - U_k U_k^\top)\right].$$

Expanding:

$$E = \frac{1}{n}\text{tr}\left[X^\top X - X^\top X U_k U_k^\top - U_k U_k^\top X^\top X + U_k U_k^\top X^\top X U_k U_k^\top\right].$$

Using the linearity and cyclic property of the trace:

$$E = \frac{1}{n}\left[\text{tr}(X^\top X) - 2\text{tr}(U_k^\top X^\top X U_k) + \text{tr}(U_k^\top X^\top X U_k)\right] = \frac{1}{n}\left[\text{tr}(X^\top X) - \text{tr}(U_k^\top X^\top X U_k)\right].$$

Since $X^\top X = n\Sigma$, we have:

$$E = \text{tr}(\Sigma) - \text{tr}(U_k^\top \Sigma U_k).$$

Using $\Sigma = U\Lambda U^\top$ and orthogonality of U:

$$U_k^\top \Sigma U_k = U_k^\top U \Lambda U^\top U_k = (U_k^\top U)\Lambda(U^\top U_k).$$

Since $U_k^\top U = [I_k\ 0]$, we get:

$$U_k^\top \Sigma U_k = \Lambda_k,$$

where $\Lambda_k = \text{diag}(\lambda_1, \ldots, \lambda_k)$. Thus,

$$\text{tr}(U_k^\top \Sigma U_k) = \lambda_1 + \lambda_2 + \cdots + \lambda_k.$$

Substituting back:

$$E = \lambda_1 + \cdots + \lambda_d - (\lambda_1 + \cdots + \lambda_k) = \sum_{i=k+1}^{d} \lambda_i.$$

Hence, the final result is:

$$\frac{1}{n}\|X - \hat{X}\|_F^2 = \sum_{i=k+1}^{d} \lambda_i.$$

3. (15) Suppose you have recently been hired as a machine learning specialist at a financial services company. One of the projects assigned to you is to develop a model to identify suspicious behavior in customer data. The available dataset includes features of transactions and user profiles. Each sample also has a label indicating whether the behavior was suspicious (e.g., fraudulent) or not.

   (a) When the goal is to accurately detect suspicious behaviors, which method do you prefer between Bagging and Boosting? Justify your answer in terms of bias and variance reduction.

   (b) If your dataset contains noisy labels (e.g., incorrect class labels) or outliers, which method would you choose? Why?

   (c) Given limited computational resources and the need to deploy a model quickly, which method is a better choice in terms of training time and parallelization?

(d) One of the students has proposed the following to improve the performance of Boosting: Instead of using a single Boosting model, train several Boosting models in parallel and independently, each on a different bootstrap sample of the data. The final output is determined by majority voting among them. Conceptually, this is similar to applying Bagging on Boosting models.
Explain the advantages and disadvantages of this proposed method compared to standard Boosting, especially in the presence of noise/outliers and in terms of execution time.

Solution:

(a) When the goal is accurate detection of suspicious behavior, **Boosting** is more appropriate because it tends to reduce bias and can learn complex patterns in the data. By reducing bias, the prediction becomes more precise. In contrast, **Bagging** mainly focuses on reducing variance.

(b) In the presence of label noise or outliers, **Bagging** is a better choice because its models are trained independently, and averaging or voting helps to reduce the effect of noise. On the other hand, **Boosting** focuses on misclassified samples and may overfit by giving too much weight to erroneous data.

(c) In scenarios with limited resources and need for speed, **Bagging** is more suitable since its models can be trained independently and in parallel, leveraging parallel computation. **Boosting**, however, is inherently sequential, requiring each model to be built after the previous one, which increases training time.

(d) **Advantage:** The proposed method is more robust to noise because outliers do not appear in all models, reducing their influence in the final aggregated output.
Disadvantage: It requires more computation and resources since full Boosting must be performed multiple times, increasing execution time.

4. (15) Suppose in an ensemble method, K identical base models are independently trained on data obtained using bootstrap sampling (random sampling with replacement). The final prediction is obtained by averaging the outputs of these models.

   Prove that as the number of models K increases, the final prediction error (mean squared error) does **not worsen**.

   Use the decomposition of prediction error into three components:

   - **Bias**
   - **Variance**
   - **Irreducible Noise**

   Solution:

Bias-Variance Decomposition in Bagging

In **Bagging**, the ensemble prediction function based on averaging over K bootstrap models is defined as:

$$\bar{f}(x) = \frac{1}{K} \sum_{k=1}^{K} f_k(x)$$

The mean squared error (MSE) of the prediction is given by:

$$\text{MSE} = \mathbb{E}\left[(\bar{f}(x) - y)^2\right]$$

This error can be decomposed into the following components:

$$\text{MSE} = \underbrace{\left(\mathbb{E}[\bar{f}(x)] - \mathbb{E}[y]\right)^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}\left[\left(\bar{f}(x) - \mathbb{E}[\bar{f}(x)]\right)^2\right]}_{\text{Variance}} + \underbrace{\mathbb{E}\left[(y - \mathbb{E}[y])^2\right]}_{\text{Irreducible Noise}}$$

Now we analyze how increasing the number of models K affects each term.

1. Bias

$$\begin{aligned}
\mathbb{E}[\bar{f}(x)] &= \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} f_k(x)\right] \\
&= \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[f_k(x)] \\
&= \mathbb{E}[f_k(x)] \quad \text{(assuming all base learners are identically distributed)}
\end{aligned}$$

Thus, increasing K does not affect the bias.

2. Variance

$$\text{Var}(\bar{f}(x)) = \text{Var}\left(\frac{1}{K}\sum_{k=1}^{K} f_k(x)\right)$$

Assuming the base models are independent and identically distributed:

$$\begin{aligned}
\text{Var}(\bar{f}(x)) &= \frac{1}{K^2}\sum_{k=1}^{K} \text{Var}(f_k(x)) = \frac{1}{K^2} K \cdot \text{Var}(f_k(x)) \\
&= \frac{\text{Var}(f_k(x))}{K}
\end{aligned}$$

So the variance **decreases** as K increases.

3. Irreducible Noise

The irreducible noise is independent of the model and is defined as:

$$\sigma^2 = \mathbb{E}[(y - \mathbb{E}[y])^2]$$

Conclusion

By increasing the number of models K, the variance term decreases, while the bias and irreducible noise remain unchanged. Therefore, the overall prediction error (MSE) does not increase—and generally decreases—as K increases.

5. (15 + 15) Complete the attached notebooks. You are permitted to use chatbots or other resources for assistance, but you must ensure that you fully understand the code and implementations. During the online sessions, you may be asked to explain specific functions, code lines, and your overall approach. Be prepared to demonstrate your understanding in detail.