**TTIC 31020: Introduction to Machine Learning** **Winter 2025**

# Homework 2

Due: 3 pm, January 16th, 2025.

**Note** *You may discuss these problems in groups. However, you must write up your own solutions and mention the names of the people in your group. Also, please do mention any books, papers or other sources you refer to. We recommend that you typeset your solutions in LaTeX. Please submit your solutions as a PDF document on Canvas.*

**Challenge and Optional Questions** *Challenge questions are marked with* challenge *and optional questions are marked with* optional . *You can get extra credit for solving* challenge *questions. You are not required to turn in* optional *questions, but we encourage you to solve them for your understanding. Course staff will help with* optional *questions but will prioritize queries on non-*optional *questions.*

**Notation** In all questions, we will use the following notation: let $\mathcal{X}$ and $\mathcal{Y}$ be the instance space and label space respectively. A predictor is a mapping $h : \mathcal{X} \to \mathcal{Y}$, which outputs a label $h(x)$ for data point $x$.

1. **Nearest Neighbor Prediction in High Dimensions**

   In class we discussed *memorization*. Given a training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, memorization refers to using a $h$ such that $h(x_i) = y_i$ for each observed instance $x_i$. This does not rely on any prior information, inductive bias, or structure in the instance space $\mathcal{X}$, but does not define the behavior on unseen instances. This is very limiting. In particular, in a continuous domain $\mathcal{X}$ we might never see the same instance twice and thus memorization is futile. Even in a large discrete space, for example, discretizing images to several megapixels with a 24-bit color map, we will likely never observe the exact same image twice.

   One way of extending memorization is by letting each observed label $y_i$ be associated not only with $x_i$ but also with instances $x$ that are close to $x_i$, using the Nearest Neighbor rule which you implemented in Homework 1. The prediction of an unseen instance $x$ is based on *a* closest instance $x_i$ from the observed ones. Given a distance measure $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, define the Nearest Neighbor learning rule as follows:

   - Find a closest training point $i^* \in \arg\min_i \rho(x, x_i)$. In case of ties, pick an *arbitrary* closest point.
   - Output the label $y_{i^*}$.

   When instances are given as vectors in $\mathbb{R}^d$, a natural distance measure is the Euclidean distance $\rho(x, x') = \|x - x'\|$.

   As we shall see in this homework, the Nearest Neighbor performs well on low-dimensional data. However, it suffers from the *curse of dimensionality* even for easy learning problems. Consider $\mathcal{X} = \mathbb{R}^d$ and a situation where the true label is directly given by the sign of one of the features, but we don't know which one. This corresponds to the hypothesis class $\mathcal{H} = \{h_i(x) = \operatorname{sign}(x[i]) \mid i = 1, \ldots, d\}$, the $d$ predictors based on individual coordinates.

   (a) To establish that this is an "easy" learning problem, derive the worst-case mistake bound (in terms of $d$) for the learning rule HALVING discussed in the class.

   (b) Now consider using the Nearest Neighbor learning rule with respect to the Euclidean distance. In each step of the online learning algorithm, the learner applies the Nearest Neighbor rule to the points seen so far, and predicts the label based on one of the points seen so far that is closest to $x_i$. Show that the number of mistakes can be exponential in $d$, even if all points in the sequence have binary coordinates: $x_i \in \{\pm 1\}^d$. That is, show a sequence of labeled samples, realizable by $\mathcal{H}$, where the Nearest Neighbor learning rule may make $2^{\Omega(d)}$ mistakes.

(c) challenge optional Show a similar lower bound in the statistical setting. Consider a data distribution $\mathcal{D}$ where $x \in \mathcal{X} = \mathbb{R}^d$ are spherical Gaussian (or if you prefer: uniform on a sphere, or uniform on $\{\pm 1\}^d$) and labels are $y = \text{sign}(x[i])$ for some fixed $i$. Consider a Nearest Neighbor predictor $h_m$ based on a sequence $S \sim \mathcal{D}^m$ of $m$ i.i.d. samples from $\mathcal{D}$. Show that even with $m = 2^{cd}$ samples, for some constant $0 < c < 1$, $\mathbb{E}_S [L(h_m)] > 0.4$ (in fact, the $L(h_m) > 0.4$ with high probability). That is, $2^{\Omega(d)}$ samples are needed to ensure small error.

(d) challenge optional (i) In the same online setup and the Nearest Neighbor rule from part (b), how many mistakes can you get with real valued $\mathcal{X} = R^d$ for small $d$, even $d = 1$? (ii) Show that this is because of the worst-case (adversarial) choice of the order in which points appear in the online model, and that if examples appear in *a random order*, then the expected number of mistakes is small. Specifically, show that for any sequence of points, after permuting them with a random permutation, the expected (over the permutation) number of mistakes the online nearest neighbor rule will make is small. Alternatively, show that for any distribution $\mathcal{D}$, the expected number of mistakes for a sequence of points drawn i.i.d. from $\mathcal{D}$ will be small. (iii) Show that for $\mathcal{X} = \mathbb{R}^d$ with $d = 2$, even for the i.i.d. sequence from an adversarial distribution or on a uniformly permuted sequence, the nearest neighbor makes as many mistakes (in expectation) as the online setting (up to a constant factor).

**Discussion.** We see that in low dimensions, the problematic worst-case behavior or nearest neighbor is not indicative of typical performances on random data, or in the statistical setting. This illustrates how the online setting might be strictly more challenging than the statistical setting.

More importantly, in high dimensions, even in the statistical setting the nearest neighbor rule may require exponential in dimension many samples. We refer to this as the *curse of dimensionality*, and the Nearest Neighbor rule certainly suffers from it. Importantly, this exponential dependence is on the *intrinsic dimensionality* of the data, that is the dimensionality of the subspace or manifold on which the data lies, and not in the *ambient dimension*, i.e. the dimensionality in which it is represented. For example, perhaps images represented by vectors of millions of pixel values actually occupy only a much lower dimensional manifold of 'natural images'. See below, and [SSBD14, Section 19.2], for a discussion of learning grantees and bounds on the number of requires samples for learning using the Nearest Neighbor rule in low dimensions that establish that, with some additional assumption, $2^{O(d)}$ samples are also sufficient for learning.

**The Bayes Optimal and Parzan Predictor.** We will first discuss the Bayes Optimal Predictor (or simply "Bayes Predictor") for a joint distribution over $(\mathcal{X}, \mathcal{Y})$; then define the Parzen Window Predictor, which is the Bayes Optimal Predictor for a certain estimated joint distribution.

Recall that for a joint distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and a predictor $h$ we defined its *expected error* as:

$$L_{\mathcal{D}}(h) = \mathbf{P}_{(x,y)\sim\mathcal{D}} [h(x) \neq y] = \mathbb{E}_{(x,y)\sim\mathcal{D}} [\mathbb{1}\{h(x) \neq y\}] . \qquad (1)$$

The Bayes Error of $\mathcal{D}$ is[1]

$$L^*_{\mathcal{D}} = \inf_h L_{\mathcal{D}}(h),$$

i.e. the smallest possible error of any predictor $h : \mathcal{X} \to \mathcal{Y}$ on $\mathcal{D}$. Moreover, the infimum is also achieved with a Bayes Optimal Predictor $h_{\text{Bayes}(\mathcal{D})} : \mathcal{X} \to \mathcal{Y}$ given by

$$h_{\text{Bayes}(\mathcal{D})}(x) = \arg\max_{y\in\mathcal{Y}} \mathbf{P}_{\mathcal{D}}(Y = y|X = x) = \arg\max_{y\in\mathcal{Y}} \mathbf{P}_{\mathcal{D}}(X = x, Y = y). \qquad (2)$$

[1] If you want to be formal and very careful, the infimum is over $h$ that are *measureable* in the $\sigma$-algebra over which $\mathcal{D}$ is defined. This can be important since for an abstract space $\mathcal{X}$ (e.g. all possible views, or all possible people) this $\sigma$-algebra indicates how we observe $x \in \mathcal{X}$, or what the predictor can depend on. **But in this course we will not discuss or worry about measureability, certainly not in a formal way.**

In other words, the optimal prediction of $x$ (minimizing the probability of error) is the label most frequently associated with $x$ (breaking "ties" arbitrarily). This is because the prediction $h(x)$ of each instance $x$ is unconstrained by the prediction on any other $x' \neq x$. As claimed, the Bayes Error of $\mathcal{D}$ is $L^*_{\mathcal{D}} = L_{\mathcal{D}}\left(h_{\text{Bayes}(\mathcal{D})}\right)$.

For binary classification problems, with $\mathcal{Y} = \{-1, +1\}$, this can also be written as:

$$h_{\text{Bayes}(\mathcal{D})}(x) = \text{sign}\left(\eta_{\mathcal{D}}(x) - \frac{1}{2}\right) \tag{3}$$

where $\eta_{\mathcal{D}}(x) = \mathbf{P}_{\mathcal{D}}(Y = +1 | X = x)$ is the *posterior* probability of the label being positive after observing $x$.

**Comprehension and review questions (Do NOT turn these in)** When is the Bayes Error equal to zero? How does $\eta(x)$ look like when the Bayes Error is equal to zero? Can you write an expression for the Bayes Error in terms of $\eta(x)$ (hint: take an expectation w.r.t. $x$)? Is the Bayes Predictor unique? When is it not unique and what happens then? What happens when some $x$ are outside the support of $\mathcal{D}$ (or rather, its marginal over $X$)? Note that in this case the conditional $\mathbf{P}(Y|X = x)$, and so $\eta(x)$ are not even defined. Prove the characterization Equation (2). Prove the second equality in Equation (2) by using Bayes Rule and noting that the marginal $\mathbf{P}(x)$ doesn't depend on $y$. Show that for binary problems, Equation (2) is given by Equation (3).

If we knew, and could directly work with, the true joint distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$, we would just use the Bayes Optimal Predictor. When we do not know $\mathcal{D}$, or perhaps when it is too complicated to represent and work with, we can build an estimated $\hat{\mathcal{D}}$ based on a sample $S = \{(x_i, y_i)\}_{i=1}^m$ and then use the Bayes Optimal Predictor w.r.t. $\hat{\mathcal{D}}$.

2. **Parzen Window Predictor**

Given $S = \{(x_i, y_i)\}_{i=1}^m$, the Parzen Window (a.k.a. Kernel) Density Estimate $\hat{f}(x|Y = y)$ of the conditional densities[2] $f(x|Y = y)$, for each label $y \in \mathcal{Y}$, is defined as:

$$\hat{f}(x|y) = Z_y \sum_{i \text{ s.t. } y_i = y} K(x, x_i) \tag{4}$$

where $Z_y \in \mathbb{R}$ is a normalization factor that ensures $\int_x \hat{f}(x|y)\, dx = 1$, and $K$ is the kernel[3]

$$K(x, x_i) = e^{-\rho(x, x_i)^2/\sigma^2} \tag{5}$$

where $\rho(x, x_i)$ is some shift-invariant distance measure $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, e.g. $\rho(x, x_i) = \|x - x_i\|$ with $\mathcal{X} = \mathbb{R}^d$, and $\sigma$ is a hyper-parameter we need to manually set when using the method. The shift-invariance implies that $\int_x K(x, x_i)\, dx$ is the same constant for all $x_i$. The Parzen Window estimator $\hat{\mathcal{D}}(\mathcal{X}, \mathcal{Y})$ of the joint distribution is given by the conditional densities in Equation (4) combined with the empirical marginal $\hat{p}(y) = \frac{1}{m} |\{i | y_i = y\}|$. The empirical marginal is simply the count of different labels among the $m$ samples. The Parzen Predictor is then defined as the Bayes Optimal Predictor for $\hat{\mathcal{D}}$. Finally, the Parzen Predictor is specified by the choice of distance measure $\rho(x, x')$ and width parameter $\sigma$.

[2] If you are worried about whether it is well defined to discuss a density here, you are right. To do so we need to refer to some base measure on $\mathcal{X}$. The choice of this base measure will affect the meaning of the density, and so the resulting estimated distribution. This choice of base measure is not so obvious, especially for an abstract space $\mathcal{X}$, and could be thought of as part of our inductive bias and assumptions about the structure of $\mathcal{X}$. But this discussion is beyond the scope of this course.

[3] Beyond the Gaussian kernel used here, other kernels are also in common use. See, e.g., the Wikipedia page for "Kernel (statistics)". Many common kernels have bounded support, including the original "window", $K(x, x_i) = 1_{\rho(x, x_i) < 1}$. In this question, we use an exponentially decaying kernel to make a connection with the nearest neighbor predictor. **Question to think about:** which answers in this question would change, and how, if we use other kernels?

(a) Show that for binary $\mathcal{Y} = \{-1, +1\}$, the Parzen Predictor is given by:

$$h(x) = \operatorname{sign}\left(\sum_{i=1}^{m} y_i K\left(x, x_i\right)\right). \tag{6}$$

(b) Using the kernel $K\left(x, x'\right) = e^{-\rho(x,x')^2/\sigma^2}$, how does the Parzen predictor $h$ behave in the limit as $\sigma \to \infty$?

(c) Using the same kernel, how does $h$ behave in the limit as $\sigma \to 0$?

(d) How is the Parzen predictor related to the Nearest Neighbor predictor Question 1 when $\sigma \to 0$? Consider, in particular, the situation of "ties", i.e. where there are multiple points in $S$ that are the same distance to $x$.

**Discussion.** In Question 1, we showed that the sample complexity of Nearest Neighbor prediction is exponential in the dimension, even in very simple cases. You might want to verify that the same holds also for Parzen Window prediction with any width, or even if the width is selected based on the data.

3. **Nearest Neighbor in the Statistical Setting**

In this question we study Nearest Neighbor prediction based on a sample $S \sim \mathcal{D}^m$ of $m$ i.i.d. samples from some source distribution $\mathcal{D}(X, Y)$.

(a) Consider a source distribution where $x \in \mathcal{X} = \mathbb{R}$ is uniform on $[-1, 1] \subset \mathbb{R}$, $\mathcal{Y} = \{\pm 1\}$ and $\mathbf{P}_{\mathcal{D}}(Y = +1|x) = 0.5 + 0.3\,\operatorname{sign}(x)$ (i.e. either 0.2 or 0.8). What is the Bayes Optimal Predictor and the Bayes Error?

(b) For the above source distribution, how does the error $L\left(h_m\right)$ of the Nearest Neighbor predictor $h_m$ behave (what does it converge to) as the number of samples increases, i.e. $m \to \infty$? (Hint: when $m \to \infty$, the nearest neighbor of most $x$ in $S$ would have the same sign as $x$, but how is the label of this Nearest Neighbor distributed?)

**Guarantee on 1-Nearest Neighbor.** We see that beyond the sample complexity which is exponential in the dimension, even when the number of samples goes to infinity, the nearest neighbor predictor might have limiting error worse than the Bayes optimal. What we *can* ensure is that for a posterior $\eta_{\mathcal{D}}(x) = \mathbf{P}_{\mathcal{D}}(Y = +1 \mid X = x)$ which is $L$-Lipschitz, the expected error approaches *twice* the Bayer Error $(2L^*_{\mathcal{D}}) + \varepsilon$, with an additive error of at most $\varepsilon$, using $(L/\varepsilon)^{O(d)}$ samples, where $d$ is the intrinsic dimension [SSBD14, Section 19.2].

***k*-Nearest Neighbor.** To achieve error approaching the Bayes Error, one can instead average over several nearest neighbors, as in the $k$-**Nearest Neighbor** predictor $h$, where $h(x)$ is the majority label among the $k > 1$ points in $S$ closest to $x$.

optional Exercises 1–4 in Section 19.6 of [SSBD14], which show how, with exponential in $d$ many samples and $k$ increasing, $k$-Nearest Neighbor classification approaches the Bayes error.

**Experimentation.** In the accompanying **Jupyter Notebook**, you will experiment with $k$-Nearest Neighbor on real and synthetic data implement confidence intervals studied in Tutorial 1.

# References

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.