0 Instructions

Homework is due Thursday, February 6, 2024 at 23:59pm Central Time. Please refer to `https://courses.grainger.illinois.edu/cs446/sp2024/homework/hw/index.html` for course policy on homeworks and submission instructions.

1 Short answer: 10pts

Each question is worth 2 points. One-sentence explanations are allowed but not necessary for full credit.

1. A (1-)nearest neighbour model is trained on a dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)}\}_{i=1}^N$ where $x^{(i)} \in \mathbf{R}^d$ $\forall i$. That is, there are N training images, each of which is dimension d. It is then used to evaluate M test images. What is the time complexity of the test-time evaluation? Use big-O notation.
2. Consider two different k-nearest neighbor models: one has $k = 1$ and one with $k = 10$. In general, which would you expect to expect to have a "smoother" decision boundary?
3. Let g be the logical OR function, defined on the feature space $\{+1, -1\}^2$, which maps $g(+1, +1) = +1, g(-1, +1) = +1, g(+1, -1) = +1, g(-1, -1) = -1$. Given a linear classifier $h(x) = \text{sign}(\boldsymbol{w} \cdot \boldsymbol{x} + b)$, give a valid $(\boldsymbol{w}, b)$ pair that matches ground truth g. Let $\text{sign}(z) = +1$ for $z \geq 0$ and -1 otherwise.
4. For real matrix $A \in \mathbb{R}^{n \times m}$, what relationship does the largest singular value of A have with the largest eigenvalue of $A^\top A$?
5. As mentioned in lecture, image data does not normally satisfy the Naive Bayes assumption. Give one additional example of a real-world situation in which the Naive Bayes assumption is violated.

2 Linear Regression: 10pts

Consider a data matrix $X \in \mathbb{R}^{n \times d}$ with rows $(\boldsymbol{x}_i)_{i=1}^n$. Assume that $d > n$ and that X is full-rank; that is, $\text{rank}(X) = n$.

1. (5pts) Show that there exists a $\boldsymbol{w}$ such that the empirical risk with squared loss is zero, i.e., that $X\boldsymbol{w} = \boldsymbol{y}$.
2. (2pts) Let the SVD of X be $X = U\Sigma V^\top$. What is the rank of Σ?
3. (3pts) Show that $X^\top X$ is invertible.

3 SVM: 10 pts

1. (2pts) Recall the dual of hard-margin SVM for binary classification:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j : \boldsymbol{\alpha} \in \mathbb{R}^n, \boldsymbol{\alpha} \geq 0$$

   What is the smallest number of support vectors for a d-dimensional dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{10{,}000}$? In other words, $\boldsymbol{x}_i \in \mathbb{R}^d \, \forall i \in [n]$. Assume that $\mathcal{D}$ is linearly separable and that there exists at least one point in each class.

2. (3pts) Let an optimal solution to the dual be $\boldsymbol{\alpha} = [10, 2, 3, 0, \cdots, 0]$ (omitted elements are all 0). What are the smallest and largest number of support vectors in this case?

3. Recall the XOR problem, where we want to model the function $g_{XOR} : \{-1, 1\}^2 \rightarrow \{-1, 1\}$:

$$g_{XOR}(-1, -1) = g_{XOR}(1, 1) = 1$$
$$g_{XOR}(1, -1) = g_{XOR}(-1, 1) = -1$$

   To solve this problem, we need a nonlinear mapping. Consider the following kernel:

$$k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z} + 1)^2$$

   (a) (3pts) Write out a feature mapping $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^d$ that induces this kernel. In other words, what is one ϕ that satisfies $\phi(\boldsymbol{x})^\top \phi(\boldsymbol{z}) = k(\boldsymbol{x}, \boldsymbol{z})$?

   (b) (2pts) Find a solution $\boldsymbol{w}$ such that $h(\boldsymbol{z}) = \text{sign}(\boldsymbol{w}^\top \phi(\boldsymbol{x})) = g_{XOR}(\boldsymbol{x})$

4 Gaussian Naive Bayes: 15pts

Recall that the Bayes Classifier is

$$h(\boldsymbol{x}) = \underset{y}{\text{argmax}} \, P(y|\boldsymbol{x})$$

We will work with binary classification: $y_i \in \{-1, +1\} \, \forall i \in [n]$. The feature vectors are now continuous: $\boldsymbol{x} \in \mathbb{R}^d$.

1. (5pts) Assume that we have a prior $P(y = +1) = p$ for some $p \in (0, 1)$.

   Show that the predictor $P(y = +1|\boldsymbol{x})$ can be written can be written $\frac{1}{1+\exp\left(\log \frac{A}{B}\right)}$

where A, B are expressions in terms of $p, P(\boldsymbol{x}|y=+1), P(\boldsymbol{x}|y=-1)$.

2. (8pts) Consider a Gaussian Naive Bayes model. Let x_j be the jth element of $\boldsymbol{x}$. Let the data be generated as follows for $\boldsymbol{\mu}_+, \boldsymbol{\mu}_- \in \mathbb{R}^d$ and $I \in \mathbb{R}^{d\times d}$ the identity matrix:

$$P(\boldsymbol{x}|y=+1) = \mathcal{N}(\boldsymbol{\mu}_+, I),\ P(\boldsymbol{x}|y=-1) = \mathcal{N}(\boldsymbol{\mu}_-, I)$$

For example, the positive class has distribution

$$P(x_j|y=+1) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(x_j - \mu_{+,j})^2}{2}\right)$$

Show that the expression from the previous part, $\log \frac{A}{B}$, can be written in the form $\boldsymbol{w}^\top \boldsymbol{x} + b$, where $\boldsymbol{w}$ and b are expressions in terms of $p, \boldsymbol{\mu}_+, \boldsymbol{\mu}_-$. Identify assumptions and definitions used in your derivation.

3. (2pts) Write a single expression for $P(y|\boldsymbol{x})$ as a function of $y, \boldsymbol{x}, \boldsymbol{w}, b$.

5 Linear regression: 14pts + 1pt

Recall that the empirical risk in the linear regression method is defined as

$$\mathcal{R}(w) := \frac{1}{2n} \sum_{i=1}^{n} (w^\top x_i - y_i)^2$$

where $x_i \in \mathbb{R}^d$ is a data point and y_i is an associated label.

1. (10.5pts) **Implement** the linear regression method using gradient descent in `linear_gd(X, Y, lrate, num_iter)` function in `hw1.py`. You are given a training set `X` as input and training labels `Y` as input along with a learning rate `lrate` and maximum number of iterations `num_iter`. Using gradient descent find parameters w that minimize the empirical risk $\mathcal{R}(w)$. One iteration is equivalent to one full data gradient update step. Use a learning rate of `lrate` and only run for `num_iter` iterations. Use $w = 0$ as your initial parameters, and return your parameters w as output.

2. (3.5pts) **Implement** linear regression by setting the gradient to zero and solving for the variables, in `linear_normal(X,Y)` function in `hw1.py`. You are given a training set `X` as input and training labels `Y` as input. Return your parameters w as output.

3. (1pt) Implement the `plot_linear()` function in `hw1.py`. Use the provided function `utils.load_reg_data()` to generate a training set `X` and training labels `Y`. Plot the curve generated by `linear_normal()` along with the points from the data set. Return the plot as output. **Include the plot in your written submission.**