CS 281A - Homework 3

September 30, 2025

This assignment is due on **October 10, 2025 at 11:59PM**. Submit your solutions as a **single PDF** on bCourses. You are strongly encouraged to typeset your submission. Illegible submissions will not be graded.

Errata

- (10/06) In Problem 2.2, show with probability at least $1 - e^{-1} - e^{-2}$, not $e^{-1} + e^{-2}$.

Problem 1

Recall that for a vector $w \in \mathbb{R}^d$, $\mathcal{H}_w := \{z \ : \ \langle w, z \rangle = 0\}$.

Let $S = \{(x_i, y_i)\}$ be a set of linearly separable data in $\mathbb{R}^d$ (i.e., $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$). Define the set $\mathcal{M}_S$ to be the set of all vectors which separate the data with large dot product:

$$\mathcal{M}_S = \{w \ : \ y_i \langle w, x_i \rangle \geq 1 \quad \text{for } i = 1, \ldots, n\}\,.$$

1. Let $w_\star$ denote the element of $\mathcal{M}_S$ with smallest norm. Show that for any other w that separates the data
$$\min_{1 \leq i \leq n} \text{dist}(x_i, \mathcal{H}_w) \leq \min_{1 \leq i \leq n} \text{dist}(x_i, \mathcal{H}_{w_\star})\,.$$
2. Show that there are real numbers α_i such that $w_\star = \sum_{i=1}^n \alpha_i x_i$.
3. Show that the α_i can be chosen so that $y_i \alpha_i$ are all nonnegative.

Problem 2

Let u and v be D-dimensional unit vectors. Let M be a random matrix of dimension $d \times D$. Each entry of M is generated iid from a normal distribution with mean 0 and variance $1/d$.

1. Show that $\mathbb{E}[\langle Mu, Mv \rangle] = \langle u, v \rangle\,.$
2. Suppose $d \geq \frac{8}{\epsilon^2}$. Show that with probability at least $1 - e^{-1} - e^{-2}$,
$$\langle Mu, Mv \rangle \geq \langle u, v \rangle - \epsilon\,.$$
3. Now let's apply this to machine learning. Consider a set of n examples in D dimensional space that is linearly separable with margin γ. That is, there are n examples, (x_i, y_i) with $y_i \in \{-1, 1\}$ and $||x_i|| \leq R$, and there is a unit vetor w so that $y_i \langle w, x_i \rangle \geq \gamma$ for all i.

   Suppose that
   $$d \geq 32 \frac{R^2}{\gamma^2} \log(4n).$$
   Show that with probability at least $1/2$, $y_i \langle Mw, Mx_i \rangle \geq \frac{\gamma}{2}$ for all i. We can think of the vectors Mx_i as *embeddings* of the original data set in a lower dimensional space. This problem shows a random embedding already preserves much of the linear separability of data. An optimized embedding can do only better.

For parts 2 and 3, you can use the following fact about Gaussian random variables. If $g_1, \dots g_k$ are independent Gaussian random variables with mean zero and variance 1, then

$$\Pr\left[\frac{1}{m}\sum_{i=1}^{m} g_i^2 \geq 1+\epsilon\right] \leq \exp\left(-\frac{m\epsilon^2}{8}\right)$$
$$\Pr\left[\frac{1}{m}\sum_{i=1}^{m} g_i^2 \leq 1-\epsilon\right] \leq \exp\left(-\frac{m\epsilon^2}{4}\right).$$

Problem 3

Consider the function $k : (0,1) \times (0,1) \to \mathbb{R}$ defined by $k(x_1, x_2) = \min\{x_1, x_2\}$.

1. Prove that k is a valid kernel (Hint: write k as the integral of a product of two simple functions and then prove that its Gram matrices are positive semi-definite).

2. Now, consider a training set $\{(x_i, y_i)\}_{i=1,\dots,n}$ with $y_i \in \mathbb{R}$ and distinct points x_i in $(0,1)$. Show that if we ran kernel regression without regularization on this data set, we would obtain zero training error. More precisely, find explicit coefficients α_j, in terms of the training data, such that for all points (x_i, y_i) in the training set we have

$$\sum_{j=1}^{n} \alpha_j \min\{x_j, x_i\} = y_i.$$

Problem 4

In the high-dimensional problems, there are usually an infinite number of possible models that perfectly fit the observed data. When a problem has multiple solutions, different optimization algorithms can find entirely different solutions to the same problem. Even though all of the solutions perfectly fit the training data, their generalization performance can be vastly different. In this problem, we explore this phenomenon for two widely used optimization algorithms: gradient descent and Adam.

Consider a linear, binary classification problem under the squared loss. Let $X \in \mathbb{R}^{n \times d}$ be an $n \times d$ matrix of features, $y \in \{-1, 1\}^n$ be the corresponding vector of labels, and $\theta \in \mathbb{R}^d$ be the parameter vector. We wish to minimize the empirical risk

$$R_S[\theta] = \frac{1}{2}\|X\theta - y\|_2^2. \tag{1}$$

Assume that the rows of X are linearly independent and that $d > n$.

1. Prove that there are infinite many $\theta \in \mathbb{R}^d$ such that $R_S[\theta] = 0$.

2. Gradient descent generates a sequence of points $\left\{\theta_k^{\text{gd}}\right\}$ according to:

$$\theta_{k+1}^{\text{gd}} = \theta_k^{\text{gd}} - \alpha_k \nabla R_S[\theta_k^{\text{gd}}], \tag{2}$$

   where α_k is a fixed sequence of learning rates. Assume the sequence α_k is chosen so that gradient descent converges to a minimizer of the objective (1). (You don't need to show how to select α_k).

   Suppose we initialize $\theta_0^{\text{gd}} = 0$. Show that gradient descent converges to the minimum Euclidean norm solution to $X\theta = y$.

3. Rather than use a fixed learning rate, Adam attempts to *adapt* the learning rate for each parameter using past gradient information. In particular, Adam generates a sequence of points $\left\{\theta_k^{\text{ad}}\right\}$ according to:

$$\theta_{k+1}^{\text{ad}} = \theta_k^{\text{ad}} - \alpha_k H_k^{-1} \nabla R_S\left[\theta_k^{\text{ad}}\right] + \beta_k H_k^{-1} H_{k-1}(\theta_k^{\text{ad}} - \theta_{k-1}^{\text{ad}}), \tag{3}$$

where α_k and β_k are fixed sequences, and H_k is a positive definite, diagonal matrix

$$H_k = \text{diag}\left(\left\{\sum_{i=1}^{k} \eta_i g_i \circ g_i\right\}^{1/2}\right), \tag{4}$$

where η_k is another fixed set of coefficients, $g_k = \nabla R_S[\theta_k^{\text{ad}}]$, and $\circ$ denotes an entry-wise product. Assume the sequences $\alpha_k, \beta_k, \eta_k$ are chosen so that Adam converges to a minimizer of the objective (1). (You don't need to show how to choose these sequences).

Suppose there exists some scalar c such that $X\text{sign}(X^\top y) = cy$, and we initialize $\theta_0^{\text{ad}}, \theta_{-1}^{\text{ad}} = 0$. Prove that Adam converges to the unique solution $\theta^{\text{ad}} \propto \text{sign}(X^\top y)$.

Hint: Use induction to show every iterate satisfies $\theta_k^{\text{ad}} = \lambda_k sign(X^\top y)$ for some scalar λ_k.

4. Fix the labels $y \in \{-1, 1\}^n$, and let $X = [y; I_{n\times n}]$. Hence, only the first feature is discriminative, and the others are unrelated to the true label. Compute the solutions found by running (a) gradient descent and (b) Adam on this problem instance.

5. Compare the relative weight the solutions found in the above part place on the discriminative feature relative to the remaining features, i.e. compute $\frac{|\theta[1]|}{|\theta[i]|}$ for both gradient descent and Adam, where $\theta[i]$ denotes the i-th coordinate of θ. Heuristically, which solution do you expect to generalize better to new data?

Problem 5

Apply the concepts from the course lectures to your final project.

1. Provide your project abstract. You can copy it from the previous homework if you'd like. This may have changed from last week, and that's ok.

2. Collect a representative data set of (X, Y) pairs as you described in last week's homework. Describe what the features and labels mean for this prediction problem. Describe the geometric invariances of the data (e.g., do you expect it to behave like a sequence or an image? Or something else?)

3. Find an appropriate vector encoding of X so that you can apply linear classification. Describe why this encoding is reasonable for this set of data.

4. Fit a linear model to this data. You can use whichever software you'd like, but specify the code you used to fit this model. Is the data linearly separable? Why or why not?

5. Find a simple nonlinear lift of the data that is linearly separable. Describe the lift in equations or code. What is the associated margin? Again, you can use whichever solver you'd like to make the data separable. Be creative in constructing your lift so that it is computationally efficient to compute.