CS182: Introduction to Machine Learning
Reference Solutions of Final Exam (2023 Fall)

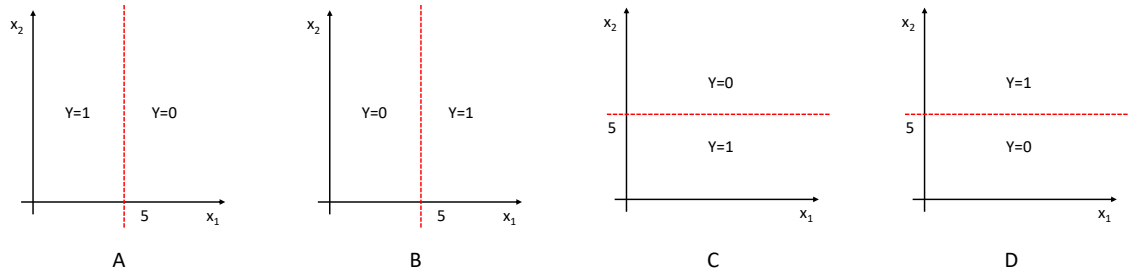
January 25, 2024

I Multiple Choice Questions

1. **[3 points]** Suppose we have a logistic regression classifier and the learned hypothesis function is

$$h_\theta(\mathbf{x}) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2),$$

where $\theta_0 = 10$, $\theta_1 = -2$, $\theta_2 = 0$ and $\sigma(\cdot)$ is the sigmoid function. Which of the following represents the decision boundary for $h_\theta(\mathbf{x})$?



Solution
(A). The decision boundary is independent of x_2. Additionally, $h_\theta(\mathbf{x})$ is smaller than 0.5 for $x_1 > 5$ (so the output is 0), and larger than 0.5 for $x_1 < 5$ (so the output is 1).

2. **[3 points]** Which of the following is a problem with the sigmoid activation function, in the context of deep neural networks?

   (a) Sigmoid is prone to vanishing gradients at extreme values.
   (b) Sigmoid can take on negative values.
   (c) Sigmoid is non-linear, which provides less representation power.
   (d) Sigmoid is numerically unstable when the input is large.

Solution
The correct answer is A, since the gradient is very small at extreme values, making gradients small. B is wrong, since sigmoid's range is in [0,1]. C is wrong because sigmoid being non-linear gives the neural network more representation power, and it's actually an advantage. D is wrong because for large inputs, the denominator approaches 1, and dividing 1 by 1 does not lead to numerical instability.

3. **[3 points]** The general form of regularized linear regression is

$$\min_{\mathbf{w}} \|(y - A\mathbf{w})\|_2^2 + \lambda \cdot r(W)$$

where $r : \mathbb{R}^d \to \mathbb{R}$ is the regularization function. Which of these regularization functions would tend to induce a sparse w ? In other words, which function would most likely cause some coefficients of w to be set to zero?

   (a) 1.
   (b) $\sum_{i=1}^{d} |w_i|$.
   (c) $\sum_{i=1}^{d} w_i^2$.
   (d) $\max(|w_i|, \ldots, |w_i^d|)$.

Solution
The correct answer is (b), which is L-1/Lasso regularization. Choice (a) adds a constant term to the arg max and, thus, does not affect the result. Choice (c) is L-2/Ridge regularization and does not induce sparsity. Choice (d) penalizes by the max coefficient, and, thus, has no effect on non-max coefficients.

4. **[3 points]** Which of the following statements about AdaBoost is (are) correct.

(a) "Ada" stands for "adaptive," as the meta learner adapts to the performance of its learners

(b) AdaBoost works best with support vector machines

(c) At test/classification time, AdaBoost computes a weighted sum of predictions

(d) AdaBoost can transform any set of classifiers to a classifier with better training accuracy

Solution
The correct answer is A, C.
B is false: linear classifiers are less well suited to ensembling than most nonlinear classifiers are. D is false: if the weak classifiers all have 50% accuracy, then AdaBoost can't build a reliable meta-learner, even for just the training points.

5. **[3 points]** Which of the following is (are) typical benefits of ensemble learning in its basic form (that is, not AdaBoost and not with randomized decision boundaries), with all weak learners having the same learning algorithm and an equal vote?

(a) Ensemble learning tends to reduce the bias of your classification algorithm.

(b) Ensemble learning tends to reduce the variance of your classification algorithm.

(c) Ensemble learning can be used to avoid overfitting.

(d) Ensemble learning can be used to avoid underfitting.

Solution
The correct answer is B, C.
In ensemble learning, increasing the number of classifiers reduces the variance of our model but generally has little effect on the bias. Therefore, basic ensembling can be used to avoid overfitting but would generally not be used to avoid underfitting. (By contrast, AdaBoost reduces bias.)

6. **[3 points]** Which of the following is **not** data generation model?

(a) Generative Adversarial Networks.

(b) Denoising Diffusion Probabilistic Models.

(c) Variational Auto-Encoder.

(d) ResNet.

Solution
The correct answer is D. Should be very obvious.

7. **[3 point]** Use y for label and $f(x)$ for prediction. The form of the BCE loss function (loss function for binary classification problems) is:

Solution
The form of the BCE loss is $\underline{-(y\log(f(x)) + (1-y)\log(1-p(x)))}$

II Naive Bayes [? points]

Based on training data in Table 1, you should construct a Naive Bayes Classifier and classifying a test document into the categories *China* (c) and *Not China* ($\neg c$). We estimate the prior probability of c by $P(c) = \frac{N(c)}{N}$, where $N(c)$ is the number of training samples belonging to c, and N is the number of training examples. A very natural way to represent words is word frequency $f(w, c)$. It is defined in the way:

$$f(w, c) = \text{number of times word, } w, \text{ appeared for category, } c.$$

Using this definition, we can then obtain the likelihood

$$P(w \mid c) = \frac{f(w, c)}{n(c)},$$

where $n(c)$ is the number of words in the training set for category c. To prevent $P(w \mid c)$ from being 0, we add smoothing to the calculation of $P(w \mid c)$; thus, we have

$$P(w \mid c) = \frac{f(w, c) + 1}{n(c) + \text{the number of unique words in the training set}}. \tag{1}$$

Note: You need not estimate parameters that you don't need for classifying the test document.

	docID	words in document	in China?
training set	1	Guangzhou Beijing	yes
	2	Macao Beijing Shanghai	yes
	3	Japan Sapporo	no
	4	Sapporo Osaka Beijing	no
	5	Guangzhou Beijing Guangzhou	yes
test set	6	Guangzhou Guangzhou Sapporo	?

Table 1: All data for Naive Bayes classifier

1. **[4 points]** Based on training data in table 1 and (1), what is the prior probabilities $P(c)$ and $P(\neg c)$? Calculate the likelihoods $P(\text{Guangzhou} \mid c)$, $P(\text{Sapporo} \mid c)$, $P(\text{Guangzhou} \mid \neg c)$ and $P(\text{Sapporo} \mid \neg c)$.

Solution

$$P(c) = \frac{3}{5}, \quad P(\text{Guangzhou} \mid c) = \frac{3+1}{8+7} = \frac{4}{15}, \quad P(\text{Sapporo} \mid c) = \frac{0+1}{8+7} = \frac{1}{15},$$
$$P(\neg c) = \frac{2}{5}, \quad P(\text{Guangzhou} \mid \neg c) = \frac{0+1}{5+7} = \frac{1}{12}, \quad P(\text{Sapporo} \mid \neg c) = \frac{2+1}{5+7} = \frac{1}{4}.$$

2. **[4 points]** Apply the classifier to the test document and determine whether document 6 is in China. Note: You don't have to calculate the exact values of posterior probabilities.

Solution

$$P(c \mid D6) \propto P(c)P(\text{Guangzhou} \mid c)^2 P(\text{Sapporo} \mid c) = \frac{3}{5} \cdot \frac{4}{15} \cdot \frac{4}{15} \cdot \frac{1}{12}$$
$$P(\neg c \mid D6) \propto P(\neg c)p(\text{Guangzhou} \mid \neg c)^2 p(\text{Sapporo} \mid \neg c) = \frac{2}{5} \cdot \frac{1}{12} \cdot \frac{1}{12} \cdot \frac{1}{4}$$

It is easy to verify $P(c \mid D6)/P(\neg c \mid D6) > 1$. So document 6 is in China.

III Probability and Estimation [10 points]

Let us consider the ordinary linear regression problem. We model the expected value of the outcome variable y as a linear function of the input variable $\mathbf{x} \in \mathbb{R}^{p+1}$:

$$\mathbb{E}(y \mid \mathbf{x}) = \mathbf{x}^\top \beta. \tag{2}$$

Assume we have a set of data $\mathcal{D}\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ sampled i.i.d. from above model such that for some $\beta \in \mathbb{R}^{p+1}$, we have $y_i \sim \mathcal{N}(\mathbf{x}_i^\top \beta, \sigma^2)$. That is

$$p(y = y_i \mid \mathbf{x}_i, \beta, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_i - \mathbf{x}_i^\top \beta)^2}{2\sigma^2}). \tag{3}$$

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ and $\mathbf{y} = [y_1, \ldots, y_n]^\top$.

1. **[2 points]** Write down the likelihood function $p(\mathbf{y} \mid \mathbf{X}, \beta, \sigma)$ for ordinary linear regression problem.

Solution

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{X}, \beta, \sigma) &= \prod_{i=1}^n p(y_i \mid \mathbf{x}_i, \beta, \sigma) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp(-\frac{\sum_{i=1}^n (y_i - \mathbf{x}_i^\top \beta)^2}{2\sigma^2}). \end{aligned}$$

2. **[6 points]** Suppose β_i's are i.i.d., we place a Laplace prior with mean $\mu \in \mathbb{R}$ and scale parameter $b \in \mathbb{R}$ on each β_i such that

$$p(\beta_i) = \frac{1}{2b} \exp(-\frac{|\beta_i - \mu|}{b}).$$

Show that the MAP estimation of β equates the linear regression with ℓ_1 regularization.

Solution

The negative log posterior is

$$\begin{aligned} -\log p(\beta \mid \mathbf{y}, \mathbf{X}, \sigma) &\propto -\log(p(\mathbf{y} \mid \mathbf{X}, \beta, \sigma) p(\beta)) \\ &\propto \frac{\sum_{i=1}^n (y_i - \mathbf{x}_i^\top \beta)^2}{2\sigma^2} + \frac{\sum_{i=0}^p |\beta_i - \mu|}{b} + C. \end{aligned}$$

Thus, MAP estimation is equivalent to

$$\hat{\beta} = \operatorname*{argmin}_\beta \ -\log p(\beta \mid \mathbf{y}, \mathbf{X}, \sigma) = \operatorname*{argmin}_\beta \ \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \|\beta - \mu \mathbf{1}\|_1,$$

where $\lambda = \frac{2\sigma^2}{b}$, $\mathbf{1}$ is the all one vector.

3. **[2 points]** Assume $\mu = 0$, write down the constraint form of ℓ_1 regularization problem in the last question. How their hyperparameters related?

Solution

Exist t such that

$$\begin{aligned} \min_\beta \quad & \|\mathbf{y} - \mathbf{X}\beta\|^2 \\ \text{subject to} \quad & \|\beta\|_1 \le t. \end{aligned}$$

λ and t are inversely related (large λ implies small t and vice versa).

IV Optimization Problem [10 points]

Consider the general form of a quadratic programming (QP) problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \end{aligned} \tag{4}$$

where $\mathbf{Q}$ is a real symmetric $n \times n$ matrix, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^n$. The inequality in (4) is interpreted element-wise.

1. **[2 points]** Suppose $\mathbf{Q}$ is positive semi-definite. Show that the QP is a convex optimization problem, i.e., the objective function is convex, the set of $\mathbf{x} \in \mathbb{R}^n$ satisfying the constraints is a convex set.

   Solution
   The objective function is $f_0(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{c}^\top \mathbf{x}$, and Hessian of $f_0(\mathbf{x})$ is $\mathbf{H}(f_0) = \mathbf{Q}$. Since $\mathbf{Q}$ is positive semi-definite, it follows that f_0 is convex.

   Suppose $\mathbf{x}_1$ and $\mathbf{x}_2$ both satisfy the constraints. Then $\forall \alpha \in [0, 1]$, we have

   $$\mathbf{A}(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) = \alpha\mathbf{A}\mathbf{x}_1 + (1-\alpha)\mathbf{A}\mathbf{x}_2 \leq \alpha\mathbf{b} + (1-\alpha)\mathbf{b} = \mathbf{b}.$$

2. **[6 points]** Assume $\mathbf{Q}$ is invertible, write down the Lagrangian for the problem, and formulate the dual problem.

   Solution
   There are no equality constraints. Thus, the Lagrangian is

   $$L(\mathbf{x}, \lambda) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \lambda^\top(\mathbf{A}\mathbf{x} - \mathbf{b}),$$

   where $\lambda \in \mathbb{R}^m_+$. The Lagrange dual function is given by

   $$g(\lambda) = \min_{\mathbf{x}} L(\mathbf{x}, \lambda).$$

   As $\forall\lambda$, L is convex in $\mathbf{x}$, we can use the first order conditions for optimality

   $$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = \mathbf{Q}\mathbf{x} + \mathbf{c} + \mathbf{A}^\top\lambda = \mathbf{0},$$

   where $\mathbf{0}$ is the all zero vector. The above equation has solution $\mathbf{x} = -\mathbf{Q}^{-1}(\mathbf{c} + \mathbf{A}^\top\lambda)$. Substitute it in L, we obtain that

   $$g(\lambda) = -\frac{1}{2}(\mathbf{c} + \mathbf{A}^\top\lambda)^\top \mathbf{Q}^{-1}(\mathbf{c} + \mathbf{A}^\top\lambda) + \lambda^\top\mathbf{b}.$$

   Thus, the dual problem is

   $$\max_{\lambda\geq\mathbf{0}} \; -\frac{1}{2}(\mathbf{c} + \mathbf{A}^\top\lambda)^\top \mathbf{Q}^{-1}(\mathbf{c} + \mathbf{A}^\top\lambda) + \lambda^\top\mathbf{b}.$$

3. **[2 points]** State the KKT conditions that the optimal solution have to satisfy.

   Solution
   Let $\mathbf{x}^*$ and λ^* be primal and dual optimal. The KKT conditions are

   - Primal feasible: $\mathbf{A}\mathbf{x}^* - \mathbf{b} \leq \mathbf{0}$.
   - Dual feasible: $\lambda^* \geq \mathbf{0}$.
   - Complementary Slackness: $\lambda^{*\top}(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = 0$.
   - Stationarity: $\mathbf{Q}\mathbf{x}^* + \mathbf{c} + \mathbf{A}^\top\lambda^* = \mathbf{0}$.

V Principal Components Analysis [11 points]

Consider the following design matrix, representing four sample points $X_i \in \mathbb{R}^2$.

$$X = \begin{bmatrix} 4 & 1 \\ 2 & 3 \\ 5 & 4 \\ 1 & 0 \end{bmatrix}.$$

We want to represent the data in only one dimension, so we turn to principal components analysis (PCA).

1. **[5 points]** Compute the **unit-length principal component directions** of X, and **state which one the PCA algorithm would choose** if you request just one principal component. Please provide an exact answer, without approximation. (You will need to use the square root symbol.) **Show your work!**

Solution

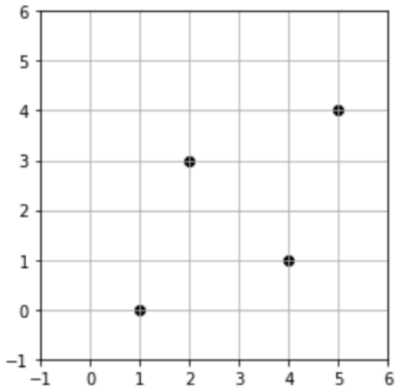
We center X, yielding

$$\dot{X} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 2 & 2 \\ -2 & -2 \end{bmatrix}$$

Then $\dot{X}^\top \dot{X} = \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix}$. (Divide by 4 if you want the sample covariance matrix. But we don't care about the magnitude.) Its eigenvectors are $\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}^\top$ with eigenvalue 16 and $\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}^\top$ with eigenvalue 4 . The former eigenvector is chosen. (Negated versions of these vectors also get full points.)

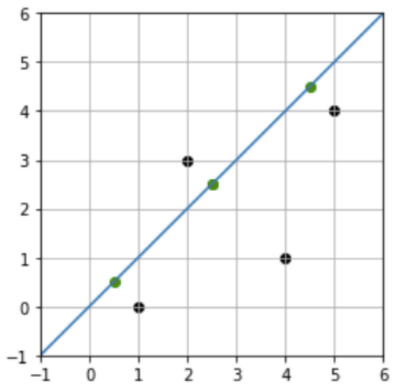
2. **[3 points]** The plot below depicts the sample points from X. We want a one-dimensional representation of the data, so **draw the principal component direction (as a line) and the projections of all four sample points onto the principal direction.**

   Label each projected point with its principal coordinate value (where the origin's principal coordinate is zero). Give the principal coordinate values exactly



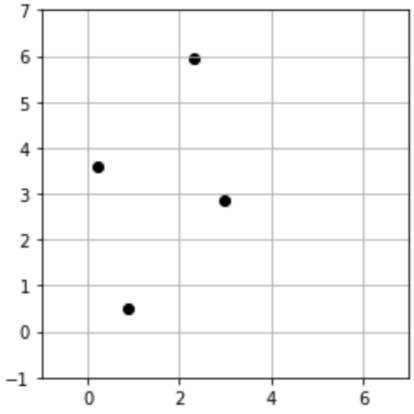
Solution

The principal coordinates are $1/\sqrt{2}, 5/\sqrt{2}, 5/\sqrt{2}, 9/\sqrt{2}$. (Alternatively, all of these could be negative, but they all have to have the same sign.)

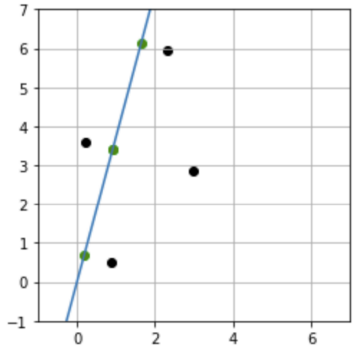


3. **[3 points]** The plot below depicts the sample points from X rotated 30 degrees counterclockwise about the origin. As in part (2), **identify the principal component direction that the PCA algorithm would choose and draw it (as a line) on the plot.** Also **draw the projections of the rotated points onto the principal direction.**

   Label each projected point with the exact value of its principal coordinate.



Solution



The line passes through the origin and is parallel to the two sample points that are farthest apart, so it's easy to draw. Rotation has not changed the principal coordinates: $1/\sqrt{2}, 5/\sqrt{2}, 5/\sqrt{2}, 9/\sqrt{2}$. (Again, these could all be negative.)

VI Stochastic gradient descent [9 points]

In this problem, we will walk through a simple example for stochastic gradient descent. Consider the quadratic loss function

$$L\left(w, \{y_i\}_{i=1}^n\right) = \frac{1}{n}\sum_{i=1}^{n} L\left(w, y_i\right) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(w - y_i\right)^2$$

where $y_1, \cdots, y_n$ are given scalars and $w \in \mathbb{R}$ is a single parameter that we wish to estimate. In order to estimate the optimal w^* that minimizes our loss function we will use stochastic gradient descent (SGD). Typically, when using SGD, we randomly sample a data point y_i from the set of all data points $\{y_i\}_{i=1}^n$ and compute the gradient update step on the sampled y_i. Rather than sampling with replacement, we consider a variant of SGD where we sample without replacement. Specifically, we will shuffle our dataset and then run SGD where we use y_1 as the sample for the time step $t = 1, y_2$ as the sample for the time step $t = 2$ and so on. For this question you may assume we have a already shuffled dataset $y_1, \cdots, y_n$.

1. **[3 points]** What is the stochastic gradient update step for w at time t under the optimization objective to minimize $L(w)$? Assume the step size is η. Write down an expression for $w^{(t+1)}$ using $\eta, w^{(t)}$ and y_t. Remember that stochastic gradient descent only considers one data point in each update step.

   Solution
   $\nabla L(w) = w^{(t)} - y_t$ for the update step t, and therefore

   $$w^{(t+1)} = w^{(t)} - \eta\left(w^{(t)} - y_t\right)$$

2. **[4 points]** Now suppose we use the dynamic step size $\eta = \frac{1}{t}$ that changes according to t. Write the expression for $w^{(t)}$ after t steps of SGD. Assume we start with $w^{(1)} = 0$. Your expression for $w^{(t)}$ should only include $y_1, \cdots, y_t$ and t. You may use scratch paper if needed, but ensure that your final answer is in the space provided below on this page.

   Solution

   $$w^{(2)} = w^{(1)} - w^{(1)} + y_1 = y_1$$
   $$w^{(3)} = w^{(2)} - \frac{1}{2}\left(w^{(2)} - y_2\right) = \frac{1}{2}\left(y_1 + y_2\right)$$

   By induction,

   $$w^{(t+1)} = w^{(t)} - \frac{1}{t}\left(w^{(t)} - y_t\right) = \frac{1}{t}\left(y_1 + \cdots + y_t\right).$$

3. **[2 points]** Instead of SGD, we can also determine a closed form solution to our optimization problem

   $$w^* = \arg\min_w L(w)$$

   What is this closed form solution? You are not required to show the derivation, though you may choose to in order to receive partial credit if your final answer is incorrect. Is the closed form solution the same as the result of running SGD for n steps (one epoch on the dataset)? If not, how does it differ?

   Solution

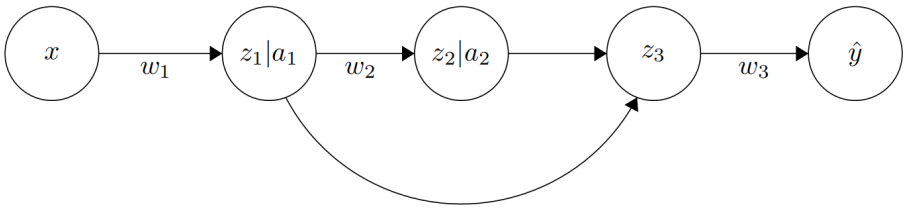
   $$\frac{d}{dw}L(w)\Big|_{w=w^*} = \frac{1}{n}\sum_{i=1}^{n}\left(w^* - y_i\right) = 0 \Longrightarrow w^* = \frac{1}{n}\sum_{i=1}^{n} y_i$$

   They are the same.

VII Neural Networks [20 points]

Residual Neural Network

Consider the following neural network, which operates on scalars.



In this network, w_1, w_2, and w_3 are scalars. The network takes the scalar x as input, and computes $z_1 = w_1 x$. The ReLU nonlinearity is then applied: $a_1 = \text{ReLU}(z_1)$. Next, the network computes $z_2 = w_2 a_1$ and applies the ReLU nonlinearity: $a_2 = \text{ReLU}(z_2)$. Next, $z_3 = a_1 + a_2$. Finally, $\hat{y} = w_3 z_3$.

We will use the mean squared error loss function $L = \frac{1}{2}(y - \hat{y})^2$ to train this network. You may use $R(x)$ and $R'(x)$ to denote ReLU and the derivative of ReLU respectively.

1. **[5 points]** Find $\frac{dL}{dw_3}$.

Solution

$$\frac{dL}{dw_3} = -(y - \hat{y}) z_3$$

2. **[5 points]** Find $\frac{dL}{dw_2}$.

Solution

$$\frac{dL}{dw_2} = -(y - \hat{y}) w_3 \frac{dR(w_2 a_1)}{dw_2} = -(y - \hat{y}) w_3 R'(z_2) a_1$$

3. **[5 points]** Find $\frac{dL}{dw_1}$.

Solution

$$\begin{aligned}\frac{dL}{dw_1} &= -(y - \hat{y}) w_3 \frac{d(R(w_1 x) + R(w_2 R(w_1 x)))}{dw_1} \\ &= -(y - \hat{y}) w_3 (R'(w_1 x) x + R'(w_2 R(w_1 x)) w_2 R'(w_1 x) x) \\ &= -(y - \hat{y}) w_3 R'(z_1) x (1 + R'(z_2) w_2)\end{aligned}$$

4. **[5 points]** How would the network change if we added a ReLU nonlinearity to unit z_3 such that $a_3 = \text{ReLU}(z_3), \hat{y} = w_3 a_3$. Briefly explain your reasoning.

Solution

We already have that $a_1 \geq 0, a_2 \geq 0$, so it would be redundant to apply ReLU to z_3.

VIII Mixture of Gaussians [? points]

A data point (x, y) in a two-dimensional real space is modeled by a probability distribution with a probability density function $f(x, y)$. Here, the probability distribution is a mixture of two distributions corresponding to class A and class B that respectively have the following probability density functions:

$$f_{\mathrm{A}}(x, y) = \mathbb{P}(x, y \mid z = A) = \frac{1}{\pi} \exp\left\{-\frac{x^2}{a} - ay^2\right\}, \tag{5}$$

$$f_{\mathrm{B}}(x, y) = \mathbb{P}(x, y \mid z = B) = \frac{1}{\pi} \exp\left\{-(x-2)^2 - (y-3)^2\right\}. \tag{6}$$

In addition, the prior probabilities (i.e., the mixture weights) of class A and class B are respectively given as

$$\mathbb{P}(z = A) = \frac{1}{1 + \exp(b)}, \quad \mathbb{P}(z = B) = 1 - \mathbb{P}(z = A). \tag{7}$$

Note that a is a positive real parameter and that b is a real parameter.

1. Answer the maximum likelihood estimate of a, when three data points, $(1, 1), (2, 2)$, and $(0, 1)$, that are known to belong to class A, are given.
2. Assume that $a = 1$. We determine whether a data point (x, y) belongs to class A or class B by comparing the posterior class probabilities. Answer the conditions for determining that the data point (x, y) belongs to class A.
3. Find the value of a when the posterior probability that data point $(1, 1)$ belongs to class A is equal to the prior probability of class A.
4. Assume that $a = 0.5$. Answer the maximum likelihood estimate of b when the two data points, $(0, 0)$ and $(1, 2)$, are observed. Note that $\exp(-10) \approx 0$ may be used.

Solution

1. The likelihood is as follows:
$$L(a) = f(1, 1) \cdot f(2, 2) \cdot f(0, 1), \tag{8}$$
The data points are known to belong to class A, therefore $f(1, 1) = f_A(1, 1), f(2, 2) = f_A(2, 2), f(0, 1) = f_A(0, 1)$. a is estimated by:
$$\hat{a} = \arg\max_a L(a). \tag{9}$$
The negative log-likelihood is given as
$$-\log L(a) = -\log\left\{\frac{1}{\pi} e^{-\frac{1}{a} - a} \cdot \frac{1}{\pi} e^{-\frac{4}{a} - 4a} \cdot \frac{1}{\pi} e^{-a}\right\} = 3\log\pi + \left(\frac{5}{a} + 6a\right). \tag{10}$$
The estimate $\hat{a}$ is derived by:
$$\hat{a} = \arg\min_a \{-\log L(a)\}, \tag{11}$$
Let $\frac{\partial\{-\log L(a)\}}{\partial a} = 0$, the closed form of $\hat{a}$ is calculated as follows:
$$\frac{5}{\hat{a}^2} - 6 = 0, \tag{12}$$
$$\hat{a} = \sqrt{30}. \tag{13}$$
2. The posterior class probability of A is given as follows:
$$\mathbb{P}(z = A \mid x, y) = \frac{\mathbb{P}(x, y \mid z = A)}{f(x, y)}. \tag{14}$$
If the data point (x, y) belongs to class A, the following equation holds:
$$\mathbb{P}(z = A \mid x, y) > 0.5, \quad (\text{or } \mathbb{P}(z = A \mid x, y) > \mathbb{P}(z = B \mid x, y)) \tag{15}$$

which gives

$$\frac{e^{-\frac{x^2}{a}-ay^2}}{e^{-\frac{x^2}{a}-ay^2}+e^b e^{-(x-2)^2-(y-3)^2}} > 0.5, \tag{16}$$

$$e^{-x^2-y^2} > e^{b-(x-2)^2-(y-3)^2}, \tag{17}$$

$$-4x+4-6y+9 > b \tag{18}$$

3. According to the question, we have the following:

$$\mathbb{P}(z=A \mid x=1, y=1) = \mathbb{P}(z=A), \tag{19}$$

$$\Rightarrow \frac{\mathbb{P}(x=1, y=1 \mid z=A)\mathbb{P}(z=A)}{f(x=1, y=1)} = \mathbb{P}(z=A), \tag{20}$$

$$\Rightarrow \mathbb{P}(x=1, y=1 \mid z=A)\left(\mathbb{P}(z=A)-1\right) + \mathbb{P}(x=1, y=1 \mid z=B)\mathbb{P}(z=B) = 0. \tag{21}$$

This gives

$$\frac{1}{\pi}e^{-\frac{1}{a}-a} \cdot \left(-\frac{e^b}{1+e^b}\right) + \frac{1}{\pi}e^{-5} \cdot \left(\frac{e^b}{1+e^b}\right) = 0, \tag{22}$$

and

$$a + \frac{1}{a} = 5, \tag{23}$$

$$a = \frac{5+2\sqrt{5}}{2}. \tag{24}$$

4. The likelihood of b is given as follows:

$$\begin{aligned} L(b) &= f(x=0, y=0) \cdot f(x=1, y=2) \\ &= \left(\frac{1}{\pi}\frac{1}{1+e^b} + \underbrace{\frac{1}{\pi}\frac{e^b}{1+e^b}e^{-13}}_{0}\right)\left(\frac{1}{\pi}\frac{1}{1+e^b}e^{-4} + \frac{1}{\pi}\frac{e^b}{1+e^b}e^{-2}\right) \\ &= \left(\frac{1}{\pi}\frac{1}{1+e^b}\right)^2 \left(e^{-4}+e^{b-2}\right). \end{aligned} \tag{25}$$

Let

$$\frac{\partial\{-\log L(b)\}}{\partial b} = 0, \tag{26}$$

and we get

$$-2 \cdot \frac{e^b}{1+e^b} + \frac{e^{b-2}}{e^{-4}+e^{b-2}} = 0, \tag{27}$$

$$\Rightarrow \hat{b} = \log(-2e^{-2}+1). \tag{28}$$

IX Elastic net regularization [? points]

A powerful method for regularizing linear regression is called elastic net regularization, which combines ridge regression (L2 regularization) and Lasso (L1 regularization).

Observe that linear regression can be probabilistically modeled as $P\left(y^{(k)} \mid \mathbf{x}^{(k)}, \mathbf{w}, \sigma^2\right) \sim \mathcal{N}\left(\mathbf{w}^T\mathbf{x}^{(k)}, \sigma^2\right)$. This means $P\left(y^{(k)} \mid \mathbf{x}^{(k)}, \mathbf{w}, \sigma^2\right) =$

$$\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)}\right)^2}{2\sigma^2}\right)$$

It is then possible to show that ridge regression is equivalent to MAP estimation with a Gaussian prior, and Lasso is equivalent to MAP estimation with a Laplace prior.

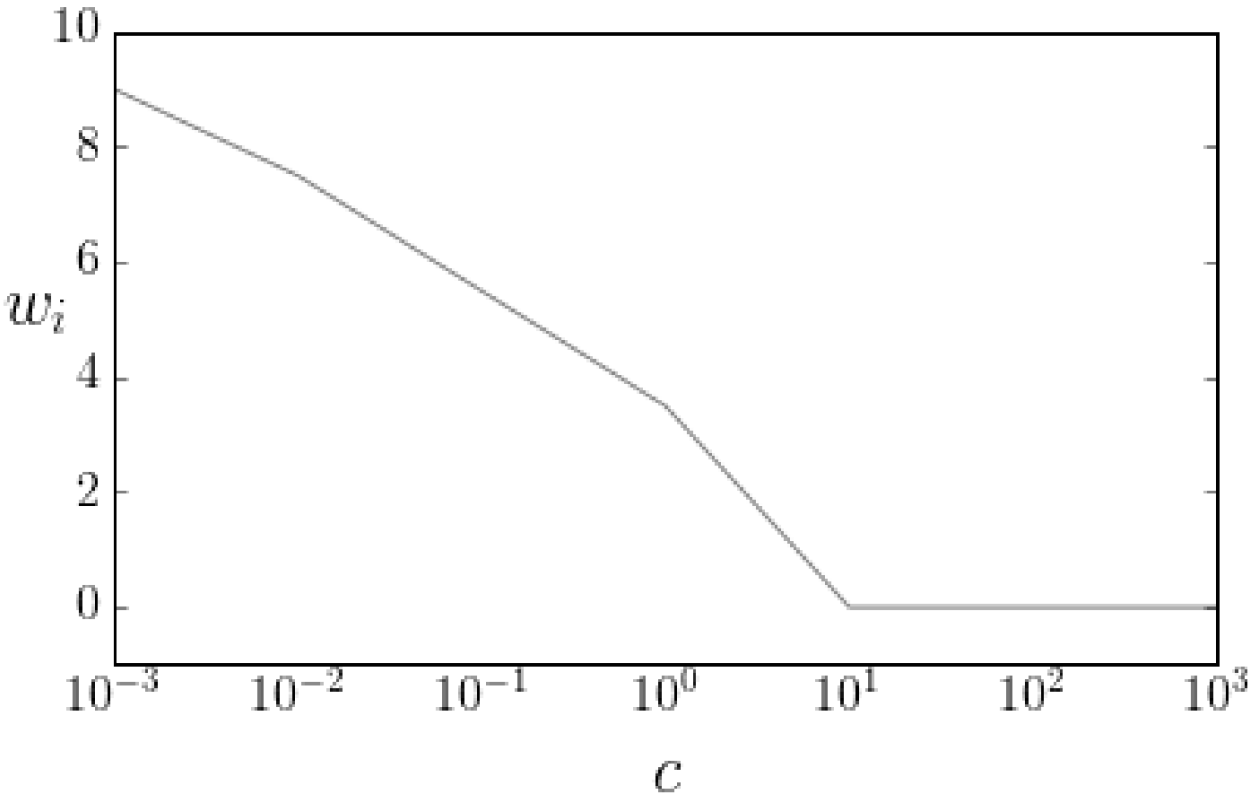
Let us assume a different prior distribution. Assume each weight w_j is i.i.d, drawn from a distribution such that $P(w_j) = q\exp\left(-\alpha_1 |w_j| - \alpha_2 w_j^2\right)$, where q, α_1, α_2 are fixed constants. Our training set is $\left(\mathbf{x}^{(k)}, y^{(k)}\right), \ldots \left(\mathbf{x}^{(n)}, y^{(n)}\right)$.

1. Show that the MAP estimate for w is equivalent to minimizing the following risk function, for some choice of constants λ_1, λ_2 :

$$R(\mathbf{w}) = \sum_{k=1}^{n}\left(y^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)}\right)^2 + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{w}\|_2^2$$

2. Suppose we scale both λ_1 and λ_2 by a positive constant c. The graph below represents the value of a single one of the weights, w_i graphed against the value of c. Out of the following set of values for λ_1, λ_2, which best corresponds to the graph? Select exactly one option and explain your choice.

A. $\lambda_1 = 1, \lambda_2 = 0$ B. $\lambda_1 = 0, \lambda_2 = 1$



Solution

1. The posterior of $\mathbf{w}$ is:

$$P\left(\mathbf{w} \mid \mathbf{x}^{(k)}, y^{(k)}\right) \propto \left(\prod_{k=1}^{n} \mathcal{N}\left(y^{(k)} \mid \mathbf{w}^T\mathbf{x}^{(k)}, \sigma^2\right)\right) \cdot P(\mathbf{w}) = \left(\prod_{k=1}^{n} \mathcal{N}\left(y^{(k)} \mid \mathbf{w}^T\mathbf{x}^{(k)}, \sigma^2\right)\right) \cdot \prod_{j=1}^{D} P\left(w_j\right)$$

Taking the log-probability, we want to maximize:

$$\begin{aligned}
l(\mathbf{w}) &= \sum_{k=1}^{n} \log \mathcal{N}\left(y^{(k)} \mid \mathbf{w}^T\mathbf{x}^{(k)}, \sigma^2\right) + \sum_{j=1}^{D} \log P\left(w_j\right) \\
&= \sum_{k=1}^{n} \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)}\right)^2}{2\sigma^2}\right)\right) + \sum_{j=1}^{D} \log q \exp\left(-\alpha_1 \left|w_j\right| - \alpha_2 w_j^2\right) \\
&= \sum_{k=1}^{n} -\frac{\left(y^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)}\right)^2}{2\sigma^2} - \alpha_1 \sum_{j=1}^{D} |w_j| - \alpha_2 \sum_{j=1}^{D} w_j^2 + n \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + D \log(q) \\
&= -\sum_{k=1}^{n} \left(y^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)}\right)^2 - 2\sigma^2\alpha_1 ||\mathbf{w}\|_1 - 2\sigma^2\alpha_2 \|\mathbf{w}\|_2^2 + n \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + D \log(q)
\end{aligned}$$

This is equivalent to minimizing the following function:

$$R(\mathbf{w}) = \sum_{k=1}^{n} \left(y^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)}\right)^2 + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2$$

where $\lambda_1 = 2\sigma^2\alpha_1, \lambda_2 = 2\sigma^2\alpha_2$.

2. B. The figure shows a regularization that induces sparsity. B is equivalent to Lasso which induces sparsity.

X Perceptron [? points]

Recall that a perceptron learns a linear classifier with weight vector $\mathbf{w}$. It predicts

$$\hat{y} = \text{sign}\left(\mathbf{w}^\intercal \mathbf{x}_t\right)$$

(assuming here that $\hat{y} \in \{+1, -1\}$. Also, note that we are not using a bias weight w_0, for simplicity). When the perceptron makes a mistake, it updates the weights using the formula

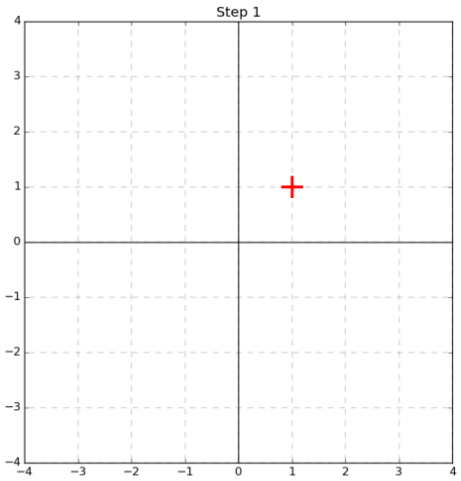
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_t \mathbf{x}_t.$$

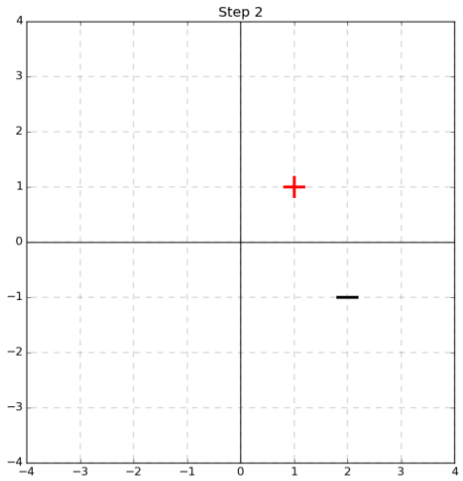
Imagine that we have $\mathbf{x}_t \in \mathbb{R}^2$, and we encounter the following data points

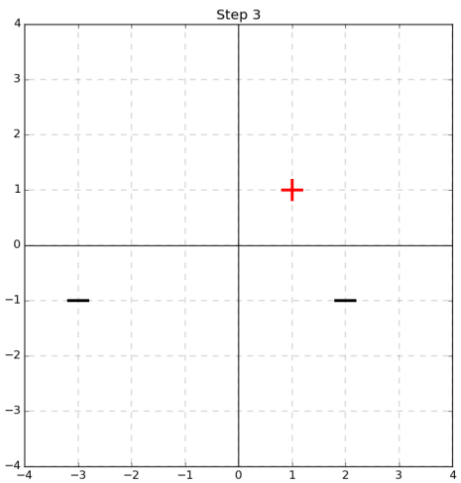
$\mathbf{x}[1]$	$\mathbf{x}[2]$	y
1	1	1
2	-1	-1
-3	-1	-1
-3	1	1

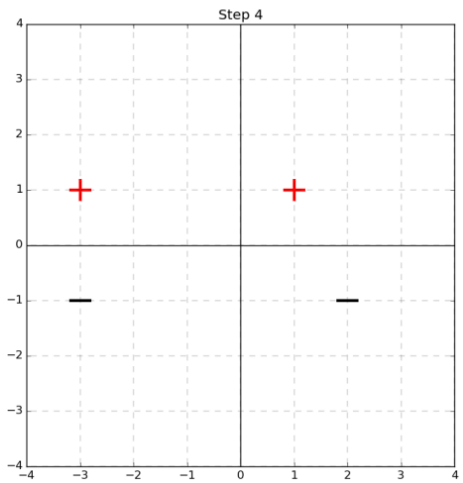
Table 2: The table of the data points.

1. Starting with $\mathbf{w}^\intercal$, use the perceptron algorithm to learn the data points in the order from top to bottom. Show the perceptron's linear decision boundary after observing each data point in the graphs below. Be sure to show which side is classified as positive.





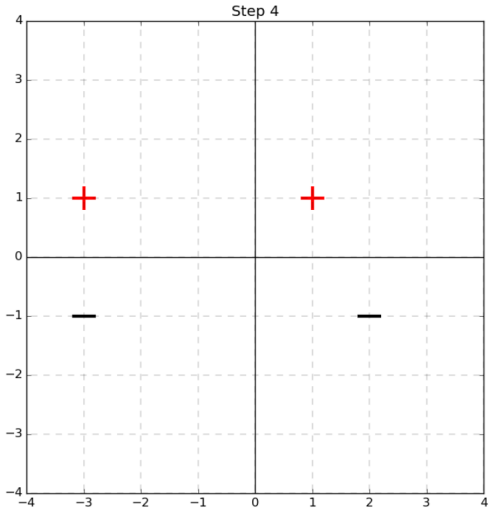




2. Does our learned perceptron maximize the margin between the training data and the decision boundary? If not, draw the maximum-margin decision boundary on the graph below.

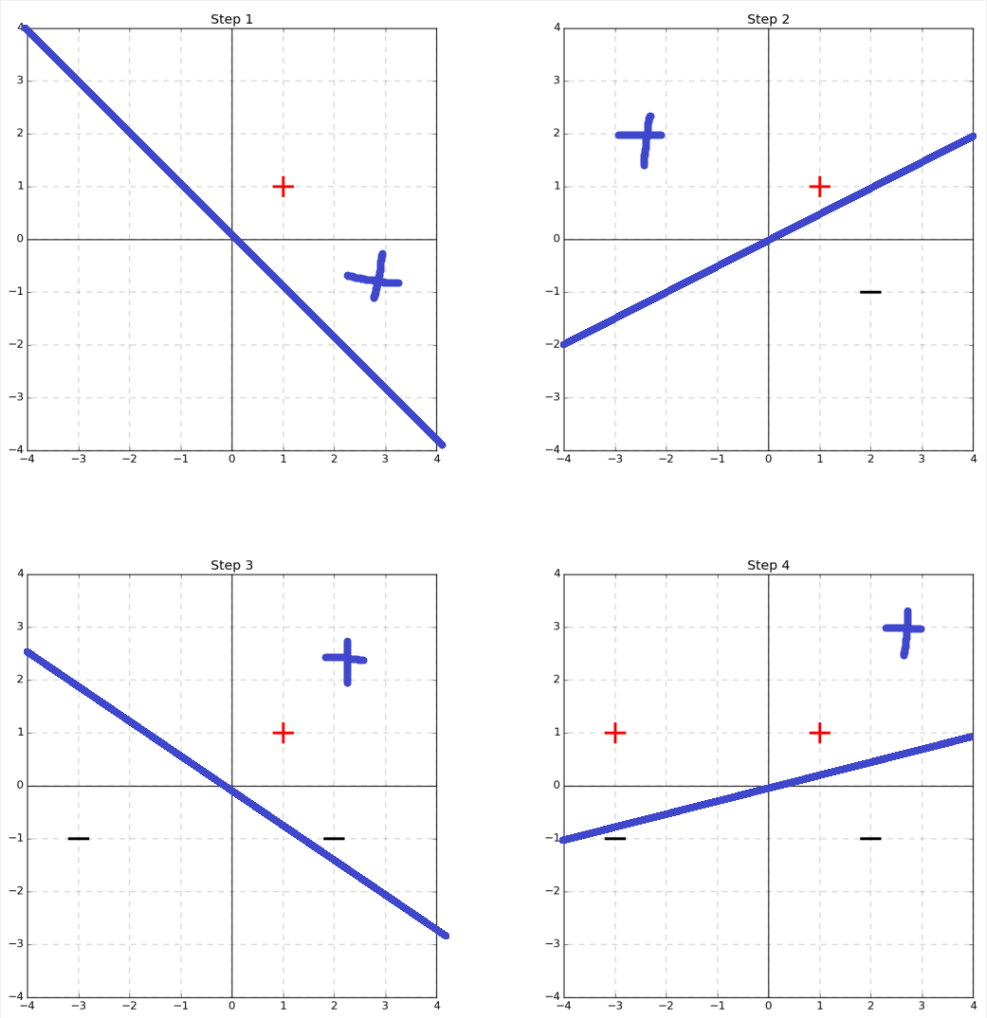
$\mathbf{x}[1]$	$\mathbf{x}[2]$	y
1	1	1
2	-1	-1
-3	-1	-1
-3	1	1

Table 3: The table of the data points. (This table is exactly the same as the table above.)

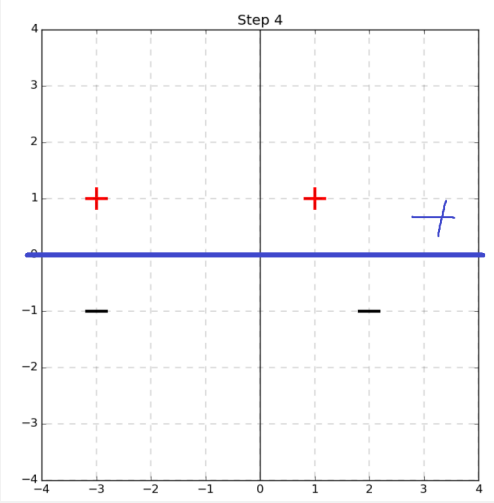


Solution

1. $(\mathbf{w}^{(1)})^\intercal = (1, 1)$, $(\mathbf{w}^{(2)})^\intercal = (-1, 2)$, $(\mathbf{w}^{(3)})^\intercal = (2, 3)$, $(\mathbf{w}^{(4)})^\intercal = (-1, 4)$.



2. No, the weight that has the maximum margin is $\mathbf{w}^\intercal = (0, 1)$.



XI Fundamentals of Machine Learning - Parameter Estimation [10 points]

Suppose a box only contains (infinitely) many red, blue, green balls. Suppose we randomly draw a ball from the box (and then put it back). The probabilities of taking a ball in different color are

$$p_r = \theta^2, \quad p_b = 2\theta(1-\theta), \quad p_g = (1-\theta)^2.$$

We repeat the process n times and observe n_r, n_b, n_g times of red, blue, green balls respectively.

1. **[2 points]** If you define $X_r = 1$ for getting a red ball, zero otherwise, then X_r is a Bernoulli random variable. Using the method of moments, find an estimator of θ.
2. **[4 points]** Following the idea of (a) and considering the proportion of blue/green balls, can you find two more estimators of θ.
3. **[4 points]** Now we have obtained three estimators $\hat{\theta}_1$, $\hat{\theta}_2$, $\hat{\theta}_3$. Use Law of Large Numbers (LLN) to show that all three $\hat{\theta}_i$ converge in probability to θ, as $n \to \infty$.

Solution

1. $$\mathbf{E}(X_r) = \theta^2$$
$$\hat{\theta}_r = \sqrt{\frac{n_r}{n}}$$

2. $$\mathbf{E}(X_b) = 2\theta(1-\theta)$$
$$\hat{\theta}_b = \frac{1 \pm \sqrt{1 - 2n_b/n}}{2}$$
$$\mathbf{E}(X_g) = (1-\theta)^2$$
$$\hat{\theta}_b = 1 - \sqrt{\frac{n_g}{n}}$$

   10

3. Given that $X_{r,i}, X_{b,i}, X_{g,i}$ are i.i.d, and $\mathbf{E}(X_r) = \theta^2, \mathbf{E}(X_b) = 2\theta(1-\theta), \mathbf{E}(X_g) = (1-\theta)^2$, hence, when $n \to \infty$, by WLLN
$$\frac{n_r}{n} = \mu_r \xrightarrow{P} \theta^2$$
$$\frac{n_g}{n} = \mu_r \xrightarrow{P} 2\theta(1-\theta)$$
$$\frac{n_b}{n} = \mu_r \xrightarrow{P} (1-\theta)^2$$
   Let $g_r(\cdot), g_g(\cdot), g_b(\cdot)$ denote the inverse mapping from μ to θ, by Theroem of SLLN,
$$\hat{\theta}_r = g_r(\frac{n_r}{n}) \xrightarrow{P} g_r(\theta^2) = \theta$$
$$\hat{\theta}_b = g_b(\frac{n_g}{n}) \xrightarrow{P} g_b(2\theta(1-\theta)) = \theta$$
$$\hat{\theta}_g = g_g(\frac{n_b}{n}) \xrightarrow{P} g_g((1-\theta)^2) = \theta$$

XII Fundamentals of Machine Learning - Probability Theory [10 points]

Prove that when randomly sampling from a normal distribution, the sample mean and the sample variance are independent.

Solution

Let $(X_i)_{n=1}^n \sim \mathcal{N}(\mu, \sigma^2)$, we can normalize it to standard normal distribution as:

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2$$

$$\begin{aligned}
&\mathbf{Conv}(\overline{X}_n, S^2) = \mathbf{E}(\overline{X}_n S^2) - \mathbf{E}(\overline{X}_n)\mathrm{E}(S^2) \\
&= \frac{1}{n(n-1)}\left(n\mathbf{E}(X^3) + n(2n-1)\mathbf{E}(X^2)\mathbf{E}(X) + n(n-1)(n-2)\mathbf{E}^3(X) + n^2\mathbf{E}(X)\mathbf{E}(\overline{X}_n^2)\right) - \mu\sigma^2 \\
&= \frac{1}{n(n-1)}\left(3n\mu\sigma^2 + n\mu(2n-1)(\sigma^2+\mu^2) + n(n-1)(n-2)\mu^3 + n^2\mu\left(\frac{\sigma^2}{n} + \mu^2\right)\right) - \mu\sigma^2 \\
&= \frac{1}{n(n-1)}\left((n^2-n)\mu\sigma^2\right) - \mu\sigma^2 \\
&= 0
\end{aligned}$$

$$\rho(\overline{X}_n, S^2) = \frac{\mathbf{Conv}(\overline{X}_n, S^2)}{\sqrt{\mathbf{Var}(\overline{X}_n)\mathbf{Var}(S^2)}} = 0$$

Hence, they are independent.

XIII Fundamentals of Machine Learning - Linear Algebra [10 points]

1. **[5 points]** If $\mathbf{A}$ is an $n \times n$ symmetric matrix all of whose eigenvalues are real and non-negative, then $\mathbf{x^T A x} \geq \mathbf{0}$ (positive semi-definite) for all nonzero $\mathbf{x}$.

2. **[5 points]** Let $\mathbf{A}$ be an invertible, real, skew-symmetric matrix ($\mathbf{A^T} = -\mathbf{A}$). Prove that $\mathbf{A^2}$ is symmetric and negative definite.

Solution

1. Let $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}]^\mathbf{T}$ From the symmetry of $\mathbf{A}$, it can be seen by eigenvalue decomposition that $\mathbf{A} = \mathbf{S^{-1}\Lambda S} = \mathbf{S^T \Lambda S}$ where $\mathbf{S}$ is orthogonal, consist of eigenvectors and D is diagonal matrix of eigenvalue. Notice that $\mathbf{x^T A x} = \mathbf{x^T S^T \Lambda S x} = (\mathbf{Sx})^\mathbf{T} \mathbf{\Lambda} (\mathbf{Sx})$. Let λ_i be the i-th eigenvalue, the quadratic is in the follow form :
$$\sum_{i=1}^{n} \lambda_i x_i^2 \geq 0$$
Hence, $\mathbf{A}$ is positive semi-definite.

2. For any $\mathbf{x} \in \mathbb{R}^\mathbf{n}$:
$$\mathbf{x^T A^2 x} = -\mathbf{x^T A^T A x} = -\langle \mathbf{Ax}, \mathbf{Ax} \rangle \leq \mathbf{0}$$
Notice that $\langle u, v \rangle = 0$ when and only when $u = 0$, and $\mathbf{A}$ is invertible so $\mathbf{Ax} = \mathbf{0}$ when and only one $\mathbf{x} = \mathbf{0}$:
$\mathbf{A^2}$ is negative definite.
Symmetric is obvious.