Discussion Section - Week 4

10/27/2020

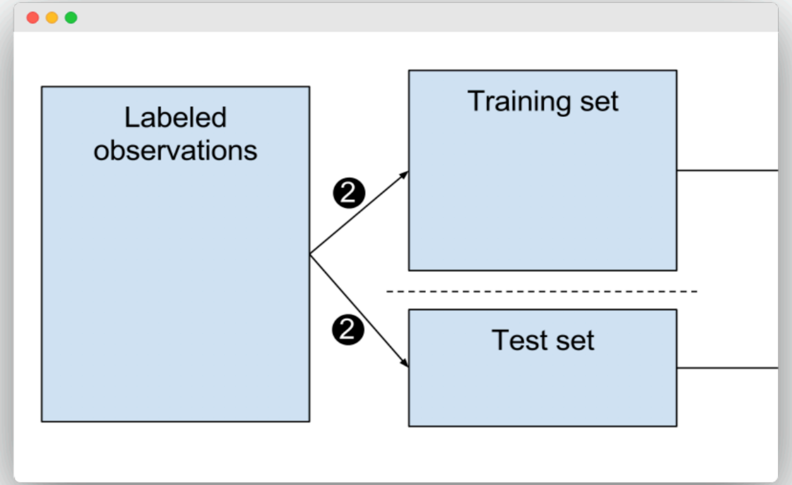
Xuehai He

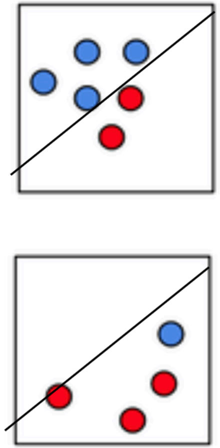
What we have learnt this week?

The exam review.
We will go though some important concepts here again.

Supervised learning, $\hat{y} =< \mathbf{w}, \mathbf{x} > +b$ such that $\hat{y} \approx y$

Data split





Math:

$$\text{Training: } S_{training} = \{(\mathbf{x}_i, y_i), i = 1..n\}$$

$$\text{Testing: } S_{testing} = \{(\mathbf{x}_i), i = 1..u\}, what\ is\ y_i?$$

One hot encoding

- One-hot encoding - to represent categorical data in a computer readable format.
  - Eg. {“Male”,“Female”}, {“Category 1”, “Category 2”,..... ,”Category N”}
  - “Male” = [1 0]
  - “Female” = [0 1]
  - “Category 2” = [0 1 0 … 0] (N elements)

Data representation

$$\mathbf{x} = \begin{array}{cc} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} & \begin{array}{l} \leftarrow \mathbf{1} \\ \leftarrow \mathbf{x} \\ \leftarrow \mathbf{x^2} \\ \\ \end{array} \end{array}$$

Feature vector

$$\mathbf{X} = \begin{array}{cl} \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} & \begin{array}{l} \leftarrow \textbf{1st data pt} \\ \leftarrow \textbf{2nd data pt} \\ \\ \\ \end{array} \end{array}$$

Design matrix

$$\mathbf{X} = \begin{array}{rc} & \begin{array}{cccc} \mathbf{1} & \mathbf{x} & \mathbf{x^2} & \\ \downarrow & \downarrow & \downarrow & \end{array} \\ \begin{array}{r} \textbf{1st data pt} \rightarrow \\ \textbf{2nd data pt} \rightarrow \\ \\ \\ \end{array} & \begin{bmatrix} x_1^{[1]} & x_2^{[1]} & \cdots & x_m^{[1]} \\ x_1^{[2]} & x_2^{[2]} & \cdots & x_m^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{[n]} & x_2^{[n]} & \cdots & x_m^{[n]} \end{bmatrix} \end{array}$$

Design matrix

Derivatives with vectors (Numerator layout)

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}^\mathsf{T} \qquad \mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\mathsf{T}$$

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \cdots & \frac{\partial y}{\partial x_n} \end{bmatrix} \qquad \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

“Jacobian formulation”

Derivatives with vectors (Denominator layout)

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}^\mathsf{T}$$

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\mathsf{T}$$

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \frac{\partial y_2}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

“Hessian formulation”

Linear regression with OLS

$$\text{Obtain/train: } f(x, W) = w_0 + w_1 x$$

$$W^* = \arg min_W \sum_i (\mathbf{x}_i^T \cdot W - y_i)^2$$

$$W = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \quad \mathbf{x}_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$$

$$W^* = \arg min_W = \arg min_W L(W) = (XW - Y)^T (XW - Y)$$

$$L(W) = W^T X^T X W - W^T X^T Y - Y^T X W + Y^T Y$$

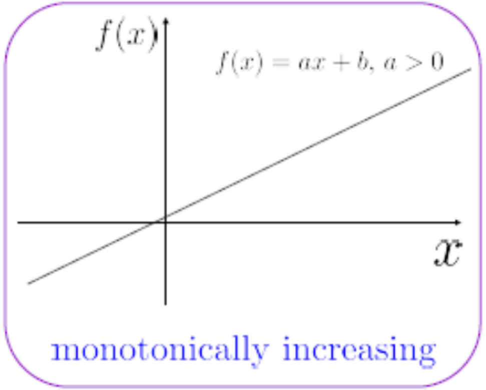
$$\frac{dL(W)}{dW} = 2X^T X W - 2X^T Y = 0$$

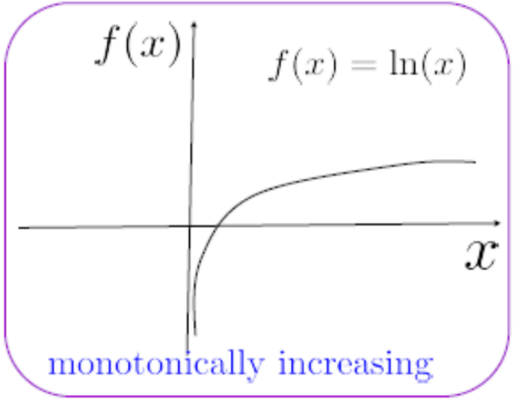
$$W^* = (X^T X)^{-1} X^T Y$$

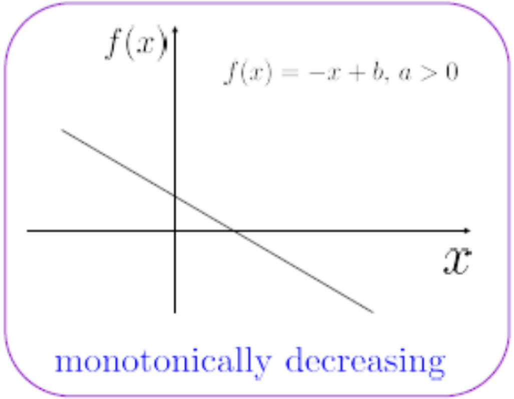
Decision boundary

- Any data sample (point) lying on the decision boundary receives a classification decision that is equally positive and negative. The decision boundary of a linear classifier is a hyper-plane. The model parameter **w** is along the normal direction of the decision boundary, pointing to the positive samples. The bias terms, b (scalar), refers to as the translation (shift) of the decision boundary.

Monotonic functions







L1 as the loss function

$$S_{training} = \{(x_i, y_i), i = 1..n\} \qquad y_i \in \mathcal{R}$$

Obtain/train: $f(x, \mathbf{w}) = w_0 + w_1 x$

$$W^* = \arg min_W \sum_{i=1}^{n} |\mathbf{x}_i^T \cdot W - y_i| \qquad W = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \quad \mathbf{x}_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$$

$$\frac{\partial |f(w)|}{\partial w} = \begin{cases} \frac{\partial f(w)}{\partial w} & f(w) > 0 \\ 0 & f(w) = 0 \\ -\frac{\partial f(w)}{\partial w} & otherwise \end{cases}$$

$$= sign(f(w)) \cdot \frac{\partial f(w)}{\partial w} \qquad sign(z) = \begin{cases} +1 & z > 0 \\ 0 & z = 0 \\ -1 & otherwise \end{cases}$$

1. Loss (Cost) Function $L(W) = \sum_{i=1}^{n} |\mathbf{x}_i^T W - y_i|$
2. Obtain the gradient $\frac{\partial L(W)}{\partial W} = \sum_{i=1}^{n} sign(\mathbf{x}_i^T W - y_i)\mathbf{x}_i$
3. Update parameter W $W_{t+1} = W_t - \lambda_t \frac{\partial L(W)}{\partial W}$

Confusion matrix and evaluation matrix

<table>
<tr><td></td><td></td><td colspan="2">True condition</td><td></td><td colspan="2"></td></tr>
<tr><td></td><td>Total population</td><td>Condition positive</td><td>Condition negative</td><td>Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$</td><td colspan="2">Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$</td></tr>
<tr><td rowspan="2">Predicted condition</td><td>Predicted condition positive</td><td>True positive</td><td>False positive, Type I error</td><td>Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$</td><td colspan="2">False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$</td></tr>
<tr><td>Predicted condition negative</td><td>False negative, Type II error</td><td>True negative</td><td>False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$</td><td colspan="2">Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$</td></tr>
<tr><td></td><td></td><td>True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$</td><td>False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$</td><td>Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$</td><td rowspan="2">Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$</td><td rowspan="2">$F_1$ score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$</td></tr>
<tr><td></td><td></td><td>False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$</td><td>Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$</td><td>Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$</td></tr>
</table>

Fall 2020 COGS 118A: Supervised Machine Learning Algorithms
Midterm Exam I Practice Problems

1 Conceptual Questions

Select the correct option(s). Note that there might be multiple correct options.

1. Choose the **most** significant difference between **regression** and **classification**:

   A. unsupervised learning vs. supervised learning.
   B. prediction of continuous values vs. prediction of class labels.
   C. features are not one-hot encoded vs features are one-hot encoded.
   D. none of the above.

   Answer: B

2. For two monotonically increasing functions $f(x)$ and $g(x)$:

   A. $f(x) + g(x)$ is always monotonically increasing.
   B. $f(x) - g(x)$ is always monotonically increasing.
   C. $f(x^2)$ is always monotonically increasing.
   D. $f(x^3)$ is always monotonically increasing.

   Answer: A D

3. For a function $f(x) = x(10 - x), x \in \mathbb{R}$, please choose the correct statement(s) below:

   A. $\arg\max_x f(x) = 5$.
   B. $\arg\min_x f(x) = 25$.
   C. $\min_x f(x) = 5$.
   D. $\max_x f(x) = 25$.

   Answer: A D

4. Assume we have a binary classification model:

$$f(\mathbf{x}) = \begin{cases} +1, & \mathbf{w} \cdot \mathbf{x} + b \geq 0, \\ -1, & \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases}$$

   where the feature vector $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, bias $b \in \mathbb{R}$, weight vector $\mathbf{w} = (w_1, w_2) \in \mathbb{R}^2$. The decision boundary of the classification model is:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

   (a) If the predictions of the classifier f and its decision boundary $\mathbf{w} \cdot \mathbf{x} + b = 0$ are shown in Figure 1, which one below can be a possible solution of weight vector $\mathbf{w}$ and bias b?

   A. $\mathbf{w} = (+1, 0), b = -1$.
   B. $\mathbf{w} = (-1, 0), b = +1$.
   C. $\mathbf{w} = (+1, 0), b = +1$.
   D. $\mathbf{w} = (0, -1), b = -1$.

   Answer: B

   (b) If the predictions of the classifier f and its decision boundary $\mathbf{w} \cdot \mathbf{x} + b = 0$ are shown in Figure 2, which one below can be a possible solution of weight vector $\mathbf{w}$ and bias b?

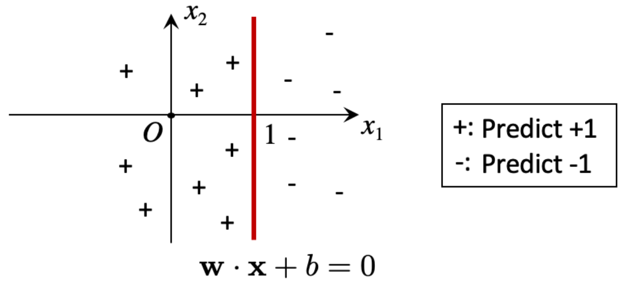


Figure 1: Decision Boundary 1

A. $\mathbf{w} = (+1, 0), b = -1$.
B. $\mathbf{w} = (-1, 0), b = +1$.
C. $\mathbf{w} = (+1, 0), b = +1$.
D. $\mathbf{w} = (0, -1), b = -1$.

Answer: C

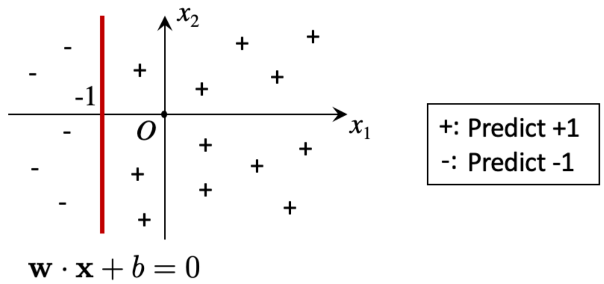


Figure 2: Decision Boundary 2

5. Suppose we have an array X with shape (150, 4), containing 150 data points where each has 4 features. We want to add one more feature (i.e. one more column) to the array X before the first feature. Which of the following option performs the task properly?

A. `X = np.vstack((np.ones((150,1)),X))`
B. `X = np.hstack((np.ones((1,150)),X))`
C. `X = np.hstack((np.ones((150,1)),X))`
D. `X = np.hstack(x,(np.ones((150,1))))`

Answer: C

6. Suppose we have a feature matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ and the corresponding ground-truth labels $Y = [y_1, y_2, \ldots, y_n]^T$. We are finding W^* that minimizes the sum of the squared error function $\mathcal{L}(W)$ by using the closed form solution to the following error function:

$$\mathcal{L}(W) = \|XW - Y\|_2^2 = (XW - Y)^T(XW - Y).$$

Assume `X` and `Y` are two `NumPy` arrays to represent the feature matrix X and the labels Y. Please write down the closed form solution to obtain the optimal W^* (i.e. `opt_W` as an array) in `NumPy` operations:

Answer: `np.dot(np.dot(np.linalg.inv(np.dot(X.T,X)),X.T),Y)`
or
`X.T.dot(X).I.dot(X.T).dot(Y)` **(works when** `X` **and** `Y` **are** `NumPy` **matrices)**

2 One-Hot Encoding

A dataset S is denoted as $S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$, where each sample $\mathbf{x}_i$ refers to the specification of a laptop computer.

	Weight (kg)	CPU **TYPE**	Component Manufacturer
$\mathbf{x}_1$	1.0	No.2	HP (100%)
$\mathbf{x}_2$	1.5	No.3	Apple (100%)
$\mathbf{x}_3$	2.0	No.1	HP (80%) and Dell (20%)
$\mathbf{x}_4$	1.5	No.2	Dell (100%)

1. Please write down a matrix in real numbers to represent the features for all the samples in dataset S. You can choose either the row vector or the column vector form to represent each sample, but please be consistent.

   Hint: Pay attention to categorical features and proportions.

   Answer:

$$\begin{bmatrix} 1.0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1.5 & 0 & 0 & 1 & 0 & 1 & 0 \\ 2.0 & 1 & 0 & 0 & 0.8 & 0 & 0.2 \\ 1.5 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Compute the **average** (in the **vector** form) of all the samples in dataset S based on the matrix you have obtained in the above question.

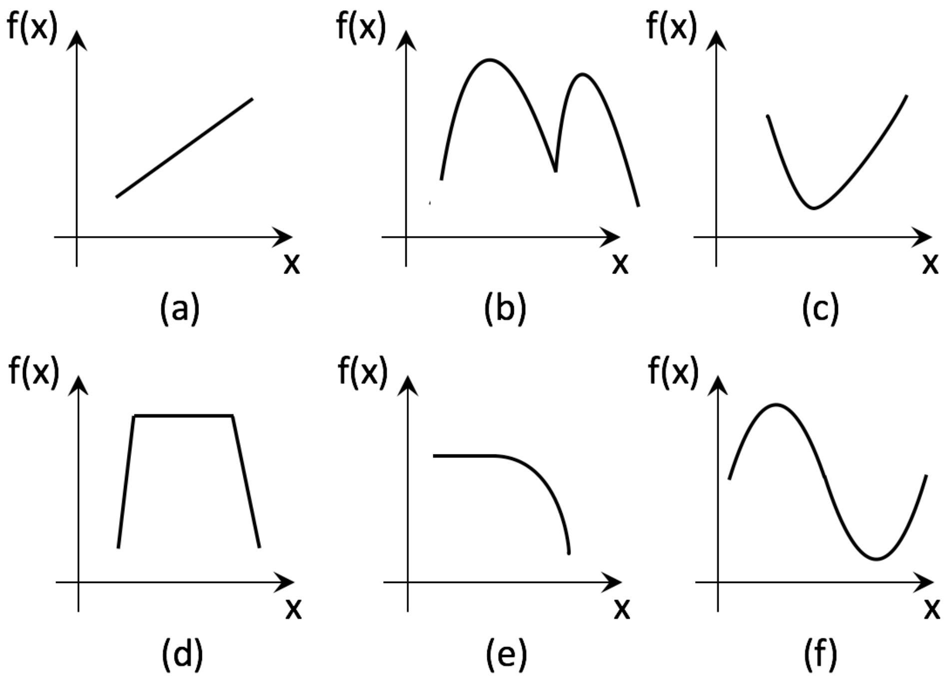
   Hint: $\frac{1}{4}\sum_{i=1}^{4} \mathbf{x}_i$ where each $\mathbf{x}_i$ is in its new feature representation above.

   Answer:

$$\begin{bmatrix} 1.5 & 0.25 & 0.5 & 0.25 & 0.45 & 0.25 & 0.3 \end{bmatrix}$$

3 Convexity I

Identify the convexity for the following six functions (a-f). Simply write down whether the function is **convex** or **non-convex** for your answers.



Answer:
Convex: a, c **Non-convex: b, d, e, f**

4 Convexity II

Give a function $f(x)$ where $x \in \mathbb{R}$. To determine the convexity of $f(x)$, we have the following rule:

$$f(x) \text{ is convex } \iff \forall x_1, x_2 \in \mathbb{R}, \forall a \in [0,1] : f(ax_1 + (1-a)x_2) \le af(x_1) + (1-a)f(x_2)$$

where " $\iff$ " means if and only if. Please use the above rule to prove that $f(x) = x^2$ is a convex function.

Answer:

$$\begin{aligned}
& af(x_1) + (1-a)f(x_2) - f(ax_1 + (1-a)x_2) \\
&= ax_1^2 + (1-a)x_2^2 - (ax_1 + (1-a)x_2)^2 \\
&= ax_1^2 + (1-a)x_2^2 - (a^2x_1^2 + 2a(1-a)x_1x_2 + (1-a)^2x_2^2) \\
&= a(1-a)x_1^2 - 2a(1-a)x_1x_2 + a(1-a)x_2^2 \\
&= a(1-a)(x_1^2 - 2x_1x_2 + x_2^2) \\
&= a(1-a)(x_1 - x_2)^2 \\
&\ge 0 \textbf{ when } a \in [0,1]
\end{aligned}$$

Thus,

$$\forall x_1, x_2 \in \mathbb{R}, \forall a \in [0,1] : f(ax_1 + (1-a)x_2) \le af(x_1) + (1-a)f(x_2)$$

which implies that $f(x) = x^2$ is a convex function.

5 Argmin and Argmax

An unknown estimator is given an estimation problem to find the minimizer and maximizer of the objective function $G(w) \in (0, 3]$:

$$(w_a, w_b) = (\arg\min_w G(w), \arg\max_w G(w)). \tag{1}$$

The solution to Eq. 1 by the estimator is $(w_a, w_b) = (15, 25)$.

Given this information, please obtain the value of w^* such that:

$$w^* = \arg\min_w [10 - 3 \times \ln(G(w))]. \tag{2}$$

Answer:

$$w^* = \arg\min_w [10 - 3 \times \ln(G(w))] = \arg\max_w G(w) = 25$$

6 Decision Boundary

(1) We are given a classifier that performs classification in $\mathbb{R}^2$ (the space of data points with 2 features (x_1, x_2)) with the following classification rule:

$$h(x_1, x_2) = \begin{cases} 1, & \text{if } \quad x_1 + 2x_2 - 4 \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Draw the decision boundary of the classifier and shade the region where the classifier predicts 1. Make sure you have marked the x_1 and x_2 axes and the intercept points on those axes.

Answer:
Please refer to HW2 Q2 solution.

(2) We are given a classifier that performs classification on $\mathbb{R}^2$ (the space of data points with 2 features (x_1, x_2)) with the following decision rule:

$$h(x_1, x_2) = \begin{cases} 1, & \text{if} \quad w_1 x_1 + w_2 x_2 + b \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Here, the normal vector $\mathbf{w}$ of the decision boundary is normalized, i.e.:

$$||\mathbf{w}||_2 = \sqrt{w_1^2 + w_2^2} = 1.$$

Compute the parameters w_1, w_2 and b for the decision boundary in Figure 3. Please make sure the predictions from the obtained classifier are consistent with Figure 3.

Hint: Please use the intercepts in the Figure 3 to find the relation between w_1, w_2 and b. Then, substitute it into the normalization constraint to solve for parameters.

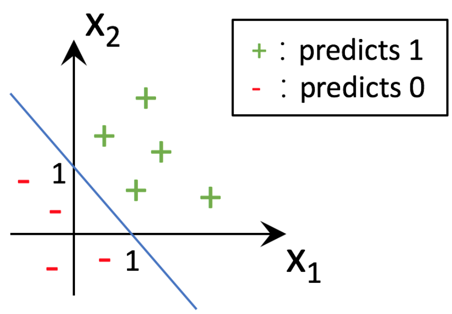


Figure 3: Decision boundary to solve for parameters.

Answer: Please refer to HW2 Q2 solution.

7 Squared Error Calculation

Assume we are given a set of points: $S = \{A(x_1 = 1, y_1 = 1), B(x_2 = 3, y_2 = 2), C(x_3 = 4, y_3 = -1), D(x_4 = 5, y_4 = 2), E(x_5 = 0, y_5 = 3)\}$ as shown in the Figure 4. In this section, we aim to fit the points in the set S with a line:

$$y = x \tag{3}$$

We define a sum-of-squares error function $\mathcal{L}$ to measure the distance between the line and points:

$$\mathcal{L} = \sum_{i=1}^{5} (y_i - x_i)^2 \tag{4}$$

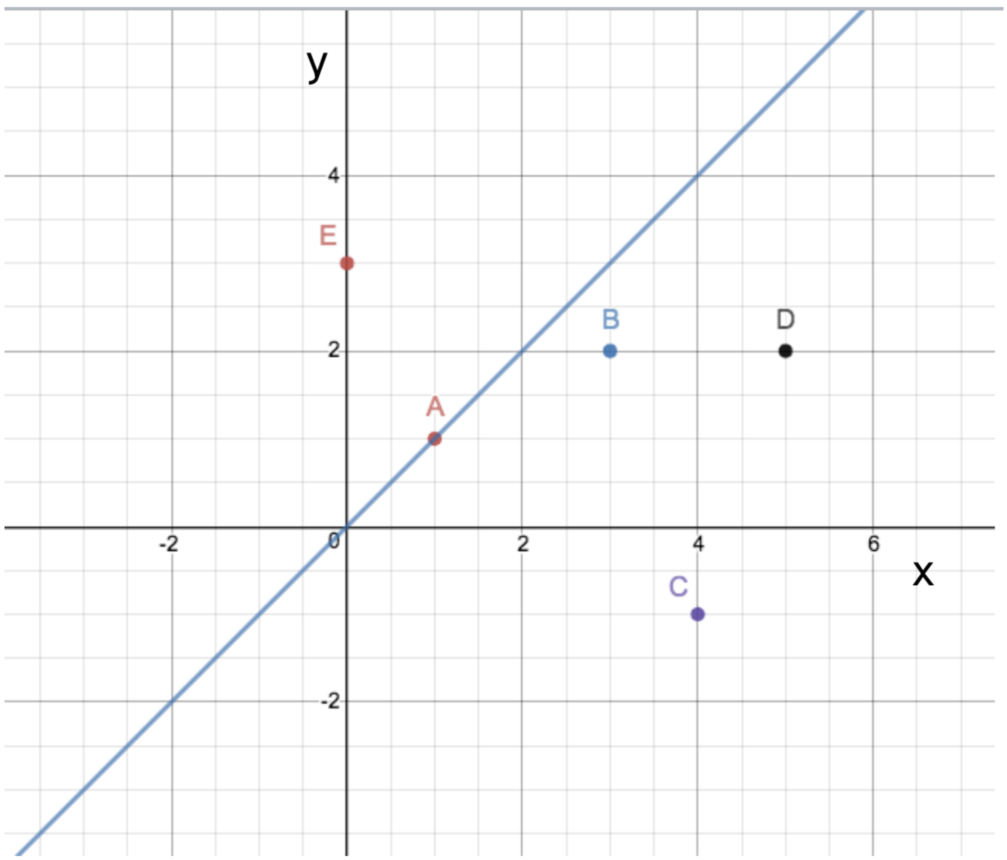


Figure 4: Line $y = x$ and points.

Please calculate the sum-of-squares error $\mathcal{L}$ according to the Figure 4.

Answer:

$$\begin{aligned}
\mathcal{L} &= \sum_{i=1}^{5} (y_i - x_i)^2 \\
&= (1-1)^2 + (2-3)^2 + ((-1)-4)^2 + (2-5)^2 + (3-0)^2 \\
&= 0 + 1 + 25 + 9 + 9 \\
&= 44
\end{aligned}$$

8 Least Squares Estimation (12 points)

Given $S = \{(x_1 = (1,2), y_1 = 3), (x_2 = (1,2), y_2 = 3), (x_3 = (1,-1), y_3 = 2)\}$, we wish to minimize:

$$\mathcal{L}(W) = \|XW - Y\|_2^2 = (XW - Y)^T (XW - Y), \tag{5}$$

where $W = [w_0, w_1, w_2]^T$. The regression function is: $y = w_0 + w_1 x + w_2 x^2$. The optimal solution to a linear regression problem is given as

$$W^* = \arg\min_W \mathcal{L}(W) = (X^T X)^{-1} X^T Y. \tag{6}$$

1. Fill in the following matrices for calculating W^* and fill in the size of all the matrices in the next line.

$$W^* = \left(\begin{bmatrix} \quad & \quad \\ & \\ & \end{bmatrix} \begin{bmatrix} \quad & \quad \\ & \\ & \end{bmatrix} \right)^{-1} \begin{bmatrix} \quad & \quad \\ & \\ & \end{bmatrix} \begin{bmatrix} \quad & \quad \\ & \\ & \end{bmatrix}$$

size: [ ] [ ] [ ] [ ]

2. What will the size of $X^T X$ be? **Answer: 2x2**

3. What will the size of $(X^T X)^{-1} X^T$ be? **Answer: 2x3**

4. What will the size of $(X^T X)^{-1} X^T Y$ be? **Answer: 2x1**

QA time

10/27/2020