Optimization and Machine Learning, Spring 2020
Homework 1
(Due Wednesday, Mar. 18 at 11:59pm (CST))

March 19, 2020

1. Suppose that we have N training samples, in which each sample is composed of p input variables and one continuous/binary response.

   (a) Please define the input and output variables, and show a linear relationship between them. (5 points)
   Solution: Define the input variable X as a vector $x \in \mathbb{R}^p$, and the output variable Y as a continuous value $y \in \mathbb{R}$ or a binary value $y \in \{0,\ 1\}$. Their linear relationship is $Y = \beta^T X$ where $\beta \in \mathbb{R}^p$ is a linear coefficient vector.

   (b) Please define a data matrix and corresponding response vector, and find your i-th ($i = 1, ..., N$) sample with its response. (5 points)
   Solution: Define a data matrix $\mathbf{X} = \begin{bmatrix} -x_1^T- \\ \vdots \\ -x_N^T- \end{bmatrix}$, and corresponding response vector $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$.
   The i-th sample is the i-th row of $\mathbf{X}$ and its response is the i-th element of $\mathbf{y}$.

   (c) Please use the least squares to estimate the parameters of the linear model in (a) based on the dataset in (b), and explain in which case the solution is unique. (10 points)
   Solution: Using the least squares, our target is to minimize

   $$RSS(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta).$$

   Differentiating with β and set the derivative to 0, we have

   $$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0.$$

   The unique solution $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ is reachable if and only if $\mathbf{X}^T\mathbf{X}$ is invertible (nonsingular, full-ranked) [Note: other reasonable explanations are acceptable].

   (d) Is there any way to get an unique closed-form solution? If yes, please show how do you obtain the solution. (5 points)
   Solution: Since there are cases that $\mathbf{X}^T\mathbf{X}$ is singular, we cannot get a unique closed-form solution. However we can add a regularization term into the original loss function, then it becomes

   $$\mathcal{L}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta,$$

   where $\lambda > 0$. Differentiating with β,

   $$\frac{\partial \mathcal{L}(\beta)}{\partial \beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) + 2\lambda\beta,$$

   Set the derivative into 0, the unique closed-form solution is $\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{y}$. It is always reachable because adding a full-ranked matrix $\lambda\mathbf{I}_p$ makes $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p$ a full-ranked matrix, and any full-ranked matrix is invertible.

   (e) How can you select the best model in (d) based only on your training data. (5 points)
   Solution: For a set of λ, we can use K-fold cross validation method. Firstly, split the dataset into random K folds, use $K-1$ folds to train the model and the rest one for validation. And then compute the average loss over the validation set using each λ candidate, finally we pick the best λ with the least average loss.

2. Given the input variables $X \in \mathbb{R}^p$ and response variable $Y \in \mathbb{R}$, the Expected Prediction Error (EPE) is defined by

$$\text{EPE}(\hat{f}) = \mathbb{E}[L(Y, \hat{f}(X))], \tag{1}$$

where $\mathbb{E}(\cdot)$ denotes the expectation over the joint distribution $\Pr(X, Y)$, and $L(Y, \hat{f}(X))$ is a loss function measuring the difference between the estimated $\hat{f}(X)$ and observed Y.

(a) Given the squared error loss $L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2$, please derive the regression function $\hat{f}(x) = \mathbb{E}(Y|X = x)$ by minimizing $\text{EPE}(\hat{f})$ w.r.t. $\hat{f}$. (5 points)
Solution: Firstly, we rewrite

$$EPE(\hat{f}) = \mathbb{E}[L(Y, \hat{f}(X))] = \mathbb{E}_X[\mathbb{E}_{Y|X}[L(Y, \hat{f}(X))|X]],$$

it is sufficient to minimize $h = \mathbb{E}_{Y|X}[L(Y, \hat{f}(X))|X] = \mathbb{E}_{Y|X}[(Y - f(X))^2|X]$, that is $\hat{f}(x) = \underset{f}{\text{argmin}}\, \mathbb{E}_{Y|X}[(Y - f)^2|X = x]$. Then we show the detail.

$$\begin{aligned}
\frac{\partial}{\partial f}\mathbb{E}_{Y|X}[(Y - f)^2|X = x] &= \frac{\partial}{\partial f}\int [y - f]^2 \Pr(y|x)dy \\
&= \int \frac{\partial}{\partial f}[y - f]^2 \Pr(y|x)dy \\
&\Rightarrow 2\int y \Pr(y|x)dy = 2f\int \Pr(y|x)dy \\
&\Rightarrow 2\mathbb{E}[Y|X = x] = 2f \\
&\Rightarrow \hat{f}(x) = \mathbb{E}[Y|X = x].
\end{aligned}$$

(b) Please explain why the nearest neighbors is an approximation to the regression function in (a). (5 points)
Solution: The nearest neighbors method $\hat{f}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i$ has two approximations. The first one is averaging over sample data to appoximate expectation, and the second one is conditioning on neighborhood to approximate conditioning on a point.

(c) Please explain how the least squares approximates the regression function in (a). (5 points)
Solution: The least square method approximates the theoretical expectation by averaging over the observed data. Using EPE in least squares, we can find the theoretical solution $\beta = \mathbb{E}(XX^T)^{-1}\mathbb{E}(XY)$, and the actual solution for least square is $\beta = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ which is an approximation for theoretical value.

(d) Please discuss the difference between the nearest neighbors and the least squares based on your results in (b) and (c). (5 points)
Solution:

- The nearest neighbors (NN) method is a locally constant function, while the least square (LS) method is a globally linear function. NN relies more on local input, while LS considers whole input.
- In terms of the number of effective parameters, LS and NN have p and N/k parameters, respectively, where N denotes the total number of training samples.
- LS usually produces high-bias and low-variance results, due to its stringent assumption on the linearity; in contrast, NN tends to make low-bias and high-variance predictions, as it has no assumption on the underlying model.

3. Given a set of observation pairs $(x_1, y_1) \cdots (x_N, y_N)$. By assuming the linear model is a reasonable approximation, we consider fitting the model via least squares approaches, in which we choose coefficients β to minimize the residual sum of squares (RSS),

$$\hat{\beta}_0,\ \hat{\beta} = \underset{\beta_0,\ \beta}{\text{argmin}}\ \sum_{i=1}^{N}(y_i - \beta_0 - \beta x_i)^2.$$

(a) Show that

$$\begin{aligned}
\hat{\beta} &= \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})^2}, \\
\hat{\beta}_0 &= \bar{y} - \hat{\beta}\bar{x},
\end{aligned} \tag{2}$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ are the sample means. (3 points)
Solution: Firstly, we compute β_0

$$\begin{aligned}
&\frac{\partial}{\partial \beta_0}\sum_{i=1}^{N}(y_1 - \beta_0 - \beta x_i)^2 = \sum_{i=1}^{N} -2(y_i - \beta_0 - \beta x_i) = 0 \\
&\Rightarrow \sum_{i=1}^{N}(y_i - \beta x_i) = \sum_{i=1}^{N}\beta_0 = N\beta_0 \\
&\Rightarrow \beta_0 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta x_i) = \frac{1}{N}\sum_{i=1}^{N} y_i - \beta\frac{1}{N}\sum_{i=1}^{N} x_i = \bar{y} - \beta\bar{x} \\
&\Rightarrow \hat{\beta}_0 = \bar{y} - \hat{\beta}\bar{x}.
\end{aligned}$$

Plug β_0 into $\sum_{i=1}^{N}(y_1 - \beta_0 - \beta x_i)^2$ and differentiate with β,

$$\begin{aligned}
&\frac{\partial}{\partial \beta}\sum_{i=1}^{N}(y_1 - \beta_0 - \beta x_i)^2 = \frac{\partial}{\partial \beta}\sum_{i=1}^{N}(y_1 - \bar{y} + \beta\bar{x} - \beta x_i)^2 = \sum_{i=1}^{N} 2[y_i - \bar{y} + \beta(\bar{x} - x_i)](\bar{x} - x_i) = 0 \\
&\Rightarrow \sum_{i=1}^{N}(y_i - \bar{y})(\bar{x} - x_i) = -\beta\sum_{i=1}^{N}(\bar{x} - x_i)^2 \\
&\Rightarrow \hat{\beta} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}.
\end{aligned}$$

In conclusion,

$$\hat{\beta} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}, \ \hat{\beta}_0 = \bar{y} - \hat{\beta}\bar{x}.$$

(b) Using (2), argue that in the case of simple linear regression, the least squares line always passes through the point $(\bar{x}, \bar{y})$. (2 points)
Solution: We can plug $(\bar{x}, \bar{y})$ into the equation $\hat{y} = \hat{\beta}x_i + \beta_0$, and we find $\bar{y} = \hat{\beta}\bar{x} + \bar{y} - \hat{\beta}\bar{x} = \bar{y}$ satisfies. So the least squares line always passes through the point $(\bar{x}, \bar{y})$.

4. Given a set of training data $(x_1, y_1), \cdots, (x_N, y_N)$ from which to estimate the parameters β, where each $x_i = [x_{i1}, \cdots, x_{ip}]^T$ denotes a vector of feature measurements for the ith sample. Consider a linear regression problem in which we want to "weight"different training examples differently. Specifically, suppose we aim at minimizing

$$\text{RSS}(\beta) = \frac{1}{2}\sum_{i=1}^{N} w_i(y_i - x_i^T\beta)^2. \tag{3}$$

(a) Show that $\text{RSS}(\beta) = (\mathbf{X}\beta - \mathbf{y})^T\mathbf{W}(\mathbf{X}\beta - \mathbf{y})$ for an appropriate diagonal matrix $\mathbf{W}$, and where $\mathbf{X} = [x_1, \cdots, x_N]^T$ and $\mathbf{y} = [y_1, \cdots, y_N]^T$. State clearly what $\mathbf{W}$ is. (1 points)
Solution: $\mathbf{W}$ is a diagonal matrix with its i-th diagonal element being $\frac{1}{2}w_i$. Suppose we have the predictions $\hat{\boldsymbol{y}} = \mathbf{X}\beta$, $\text{RSS}(\beta)$ is rewritten by

$$\text{RSS}(\beta) = (\mathbf{X}\beta - \mathbf{y})^T\mathbf{W}(\mathbf{X}\beta - \mathbf{y}) = (\hat{\mathbf{y}} - \mathbf{y})^T\mathbf{W}(\hat{\mathbf{y}} - \mathbf{y})$$

$$\begin{bmatrix} \hat{y}_1 - y_1 \\ \vdots \\ \hat{y}_N - y_N \end{bmatrix}^T \begin{pmatrix} \frac{1}{2}w_1 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{2}w_2 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}w_{N-1} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}w_N \end{pmatrix} \begin{bmatrix} \hat{y}_1 - y_1 \\ \vdots \\ \hat{y}_N - y_N \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2}w_1(\hat{y}_1 - y_1) \\ \vdots \\ \frac{1}{2}w_N(\hat{y}_N - y_N) \end{bmatrix}^T \begin{bmatrix} \hat{y}_1 - y_1 \\ \vdots \\ \hat{y}_N - y_N \end{bmatrix} = \frac{1}{2}\sum_{i=1}^{N} w_i \left(y_i - x_i^T\beta\right)^2.$$

(b) By finding the derivative $\nabla_\beta \text{RSS}(\beta)$ and setting that to zero, write the normal equations to this weighted setting and give the value of β that minimizes $\text{RSS}(\beta)$ in closed form as a function of $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{y}$. (2 points)

Solution:

$$\begin{aligned} \nabla_\beta \text{RSS}(\beta) &= \frac{\partial \,\text{RSS}(\beta)}{\partial \beta} \\ &= \frac{\partial}{\partial \beta}(\mathbf{X}\beta - \mathbf{y})^T\mathbf{W}(\mathbf{X}\beta - \mathbf{y}) \\ &= 2\mathbf{X}^T\mathbf{W}(\mathbf{X}\beta - \mathbf{y}) \\ &= 0 \\ \Rightarrow \hat{\beta} &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y}. \end{aligned}$$

(c) Suppose the y_i's were observed with differing variances. To be specific, suppose that

$$p(y_i|x_i;\beta) = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(y_i - x_i^T\beta)^2}{2\sigma_i^2}\right), \tag{4}$$

i.e., y_i has mean $x_i^T\beta$ and variance σ_i^2, where the σ_i's are fixed, known, constants). Show that finding the maximum likelihood estimate of β is equivalent to solving a weight linear regression problem. State clearly what the w_i's are in terms of the σ_i's. (4 points)

Solution: The log likelihood function is

$$\mathcal{L}(\beta) = \log\prod_{i=1}^{N} p\left(y_i|x_i;\beta\right) = \log\{\frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{\left(y_i - x_i^T\beta\right)^2}{2\sigma_i^2}\right)\} = \frac{-1}{\sqrt{2\pi}\sigma_i}\sum_{i=1}^{N}\frac{\left(y_i - x_i^\top\beta\right)^2}{2\sigma_i^2}.$$

Maximizing the likelihood is equivalent to minimizing $\sum_{i=1}^{N}\frac{\left(y_i - x_i^\top\beta\right)^2}{2\sigma_i^2}$. This is equivalent to solving a weigh linear regression problem with weight $w_i = \frac{1}{2\sigma_i^2}$.

5. To perform variable selection, three classical approaches were introduced in class, including variable subset selection, forward stepwise selection and backward stepwise selection.

(a) To deepen your understanding of these approaches, please make a table to describe their key procedures as well as the pros and cons. (6 points)

(b) Suppose we perform these three approaches on a single data set. For each approach, we obtain $p + 1$ models, containing $0, 1, 2, \cdots, p$ predictors. **Explain** your answers:

- Which of the three models with k predictors has the smallest training RSS? (1 points)
- Which of the three models with k predictors has the smallest test RSS? (1 points)

(Note that: Solutions with the correct answer but without adequate explanation will not earn credit.)

Solution:

(a) We summarize the key procedures of three approaches as follows correspondingly:

Pros:

- It is a simple and conceptually appealing approach.
- In practice, it can be instructive to observe how best-subset selection could be done for small problems.

Limitations:

- In general, there are 2^p models that involve subsets of p predictors. Consequently, best subset selection becomes computationally infeasible for values of p greater than around 40.

Pros:

- It is superior to the best subset selection in terms of computation efficiency.

Algorithm 1 Best Subset Selection

1: Let $\mathcal{M}_0$ denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.
2: **for** $k = 1, 2 \ldots, p$ **do**
3: Fit all $\begin{pmatrix} p \\ k \end{pmatrix}$ models that contain exactly k predictors.
4: Pick the best (e.g., in terms of the smallest RSS) among these $\begin{pmatrix} p \\ k \end{pmatrix}$ models, and call it $\mathcal{M}_k$.
5: **end for**
6: Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error.

Algorithm 2 Forward Stepwise Selection

1: Let $\mathcal{M}_0$ denote the null model, which contains no predictors.
2: **for** $k = 1, 2 \ldots, p-1$ **do**
3: Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
4: Choose the best (e.g., in terms of the smallest RSS) among these $p-k$ models, and call it $\mathcal{M}_{k+1}$.
5: **end for**
6: Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error.

- It can be applied even in the high-dimensional setting where $n < p$, and so is the only viable subset method when p is very large.

Limitations:

- It is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.

Pros:

- Like forward stepwise selection, the backward selection approach searches through only $1+p(p+1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection.

Limitations:

- It requires that the number of samples n is larger than the number of variables p.
- It is not guaranteed to yield the best model containing a subset of the p predictors.

(b)
- Best subset will have the smallest train RSS because the models will optimize on the training RSS and best subset will try every model that forward and backward selection will try.
- The best test RSS model could be any of the three. Best subset could easily over-fitting if the data has large p predictors relative to n observations. Forward and backward selection might not converge on the same model but try the same number of models and hard to say which selection process would be better.

6. Refer to [1, Ex. 3.5]. Consider the ridge regression problem

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \tag{5}$$

where $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage. Show that problem (5) is equivalent to the problem

$$\hat{\beta}^{\text{c}} = \underset{\beta^{\text{c}}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0^{\text{c}} - \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)\beta_j^{\text{c}})^2 + \lambda \sum_{j=1}^{p} (\beta_j^{\text{c}})^2 \right\}. \tag{6}$$

Give the correspondence between β^{c} and the original β in (5). Characterize the solution to this modified criterion. Moreover, show that a similar result holds for the least absolute shrinkage and selection operator (LASSO). (10 points)

Solution: Consider that the ridge expression problem (5) can be written as

$$\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \bar{x}_j\beta_j - \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2. \tag{7}$$

Algorithm 3 Backward Stepwise Selection

1: Let $\mathcal{M}_0$ denote the full model, which contains all p predictors.
2: **for** $k = p, p-1 \ldots, 1$ **do**
3: Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
4: Choose the best (e.g., in terms of the smallest RSS) among these k models, and call it $\mathcal{M}_{k-1}$.
5: **end for**
6: Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error.

By shifting that x_i's to have zero mean we have translated all points to the origin. As such only the 'intercept' of the data or β_0 is modified the 'slope's' or β_j^c for $i = 1, 2 \ldots, p$ are not modified. Define 'centered' values of β as

$$\beta_0^c = \beta_0 + \sum_{j=1}^{p} \bar{x}_j \beta_j$$
$$\beta_j^c = \beta_i, \quad i = 1, 2, \ldots, p,$$

that the above can be recast as

$$\sum_{i=1}^{N} \left(y_i - \beta_0^c - \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)\beta_j^c \right)^2 + \lambda \sum_{j=1}^{p} (\beta_j^c)^2.$$

The equivalence of the minimization results from the fact that if β_i minimizes its respective functional the β_i^c's will do the same. We compute the value of β_0^c in the above expression by setting the derivative with respect to this variable equal to zero (a consequence of the expression being at a minimum). We obtain

$$\sum_{i=1}^{N} \left(y_i - \beta_0^c - \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)\beta_j^c \right) = 0,$$

which implies $\beta_0^c = \frac{1}{N}\left(\sum_{i=1}^{N} y_i - \sum_{i=1}^{N} \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)\beta_j^c \right)$. Moreover, the same argument above can be used to show that the minimization required for the LASSO can be written in the same way (i.e., replace $(\beta_j^c)^2$ with $|\beta_j^c|$). The intercept in the centered case continues to be $\bar{\mathbf{y}}$.

7. It is well-known that ridge regression tends to give similar coefficient values to correlated variables, whereas the LASSO may give quite different coefficient values to correlated variables. We will now explore this property in a very simple setting.

   Suppose that $n = 2$, $p = 2$, $x_{11} = x_{12}$, $x_{21} = x_{22}$. Furthermore, suppose that $y_1 + y_2 = 0$ and $x_{11} + x_{21} = 0$ and $x_{12} + x_{22} = 0$, so that the estimate for the intercept in a least squares, ridge regression, or LASSO model is zero: $\hat{\beta}_0 = 0$.

   (a) Write out the ridge regression optimization problem in this setting. (2 points)

   (b) Argue that in this setting, the ridge coefficient estimates satisfy $\hat{\beta}_1 = \hat{\beta}_2$. (4 points)

   (c) Write out the LASSO optimization problem in this setting. (2 points)

   (d) Argue that in this setting, the LASSO coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ are not unique—in other words, there are many possible solutions to the optimization problem in (c). Describe these solutions. (2 points)

   Solution:

   (a) In this setting, the ridge regression optimization problem reads

   $$\min_{\hat{\beta}} \; f_{\text{ridge}}(\hat{\beta}) = (y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2). \tag{8}$$

   (b) It takes the following steps to obtain the solution:

   1) Expanding the equation from (8):

   $$f_{\text{ridge}}(\hat{\beta}) = (y_1^2 + \hat{\beta}_1^2 x_{11}^2 + \hat{\beta}_2^2 x_{12}^2 - 2\hat{\beta}_1 x_{11} y_1 - 2\hat{\beta}_2 x_{12} y_1 + 2\hat{\beta}_1 \hat{\beta}_2 x_{11} x_{12}) + (y_2^2 + \hat{\beta}_1^2 x_{21}^2 + \hat{\beta}_2^2 x_{22}^2 - 2\hat{\beta}_1 x_{21} y_2 - 2\hat{\beta}_2 x_{22} y_2 + 2\hat{\beta}_1 \hat{\beta}_2 x_{21} x_{22}) + \lambda \hat{\beta}_1^2 + \lambda \hat{\beta}_2^2.$$

2) Taking the partial derivative to $\hat{\beta}_1$ and setting equation to 0 to minimize:

$$\frac{\partial f_{\text{ridge}}(\hat{\beta})}{\partial \hat{\beta}_1} = (2\hat{\beta}_1 x_{11}^2 - 2x_{11}y_1 + 2\hat{\beta}_2 x_{11}x_{12}) + (2\hat{\beta}_1 x_{21}^2 - 2x_{21}y_2 + 2\hat{\beta}_2 x_{21}x_{22}) + 2\lambda\hat{\beta}_1 = 0.$$

3) Setting $x_{11} = x_{12} = x_1$ and $x_{21} = x_{22} = x_2$ and dividing both sides of the equation by 2:

$$(\hat{\beta}_1 x_1^2 - x_1 y_1 + \hat{\beta}_2 x_1^2) + (\hat{\beta}_1 x_2^2 - x_2 y_2 + \hat{\beta}_2 x_2^2) + \lambda\hat{\beta}_1 = 0,$$

$$\Downarrow$$

$$\hat{\beta}_1(x_1^2 + x_2^2) + \hat{\beta}_2(x_1^2 + x_2^2) + \lambda\hat{\beta}_1 = x_1 y_1 + x_2 y_2.$$

4) Add $2\hat{\beta}_1 x_1 x_2$ and $2\hat{\beta}_2 x_1 x_2$ to both sides of the equation:

$$\hat{\beta}_1(x_1^2 + x_2^2 + 2x_1x_2) + \hat{\beta}_2(x_1^2 + x_2^2 + 2x_1x_2) + \lambda\hat{\beta}_1 = x_1y_1 + x_2y_2 + 2\hat{\beta}_1 x_1x_2 + 2\hat{\beta}_2 x_1x_2$$

$$\Downarrow$$

$$\hat{\beta}_1(x_1 + x_2)^2 + \hat{\beta}_2(x_1 + x_2)^2 + \lambda\hat{\beta}_1 = x_1y_1 + x_2y_2 + 2\hat{\beta}_1 x_1x_2 + 2\hat{\beta}_2 x_1x_2. \tag{9}$$

5) Because $x_1 + x_2 = 0$, we can eliminate the first two terms in (9):

$$\lambda\hat{\beta}_1 = x_1y_1 + x_2y_2 + 2\hat{\beta}_1 x_1x_2 + 2\hat{\beta}_2 x_1x_2.$$

6) Similarly by taking the partial deritive to $\hat{\beta}_2$, we can get the equation:

$$\lambda\hat{\beta}_2 = x_1y_1 + x_2y_2 + 2\hat{\beta}_1 x_1x_2 + 2\hat{\beta}_2 x_1x_2.$$

7) The left side of the equations for both $\lambda\hat{\beta}_1$ and $\lambda\hat{\beta}_2$ are the same so we have:

$$\lambda\hat{\beta}_1 = \lambda\hat{\beta}_2,$$

indicating

$$\hat{\beta}_1 = \hat{\beta}_2.$$

(c) In this setting, the LASSO regression optimization problem reads

$$\min_{\hat{\beta}} \; f_{\text{LASSO}}(\hat{\beta}) = (y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(|\hat{\beta}_1| + |\hat{\beta}_2|).$$

(d) Following through the steps in (*b*), we get:

$$\lambda\frac{|\hat{\beta}_1|}{\hat{\beta}_1} = \lambda\frac{|\hat{\beta}_2|}{\hat{\beta}_2}.$$

So it seems that the LASSO just requires that $\hat{\beta}_1$ and $\hat{\beta}_2$ are both positive or both negative (ignoring possibility of 0...).

8. Refer to [1, Ex. 3.30]. Consider the elastic-net optimization problem:

$$\min_{\beta} \; \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\left[\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1\right]. \tag{10}$$

Show how one can turn this into a LASSO problem, using an augmented version of $\mathbf{X}$ and $\mathbf{y}$. (10 points)
Solution: For this problem note that if we augment $\mathbf{X}$ with a multiple of the $p \times p$ identity to get

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \gamma\mathbf{I} \end{bmatrix}, \tag{11}$$

then $\tilde{\mathbf{X}}\beta = \begin{bmatrix} \mathbf{X}\beta \\ \gamma\beta \end{bmatrix}$. If we next augment $\mathbf{y}$ with p zeros as

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}.$$

Then we have

$$\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\beta\|_2^2 = \left\| \begin{bmatrix} \mathbf{y} - \mathbf{X}\beta \\ \gamma\beta \end{bmatrix} \right\|_2^2 = \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \gamma^2\|\beta\|_2^2. \tag{12}$$

Now in the this augmented space a lasso problem for β is

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left(\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\beta\|_2^2 + \tilde{\lambda}\|\beta\|_1 \right).$$

Writing this using (12) we get in the original variables the following

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left(\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \gamma^2\|\beta\|_2^2 + \tilde{\lambda}\|\beta\|_1 \right).$$

To make this match the requested expression we take $\gamma^2 = \lambda\alpha$ and $\tilde{\lambda} = \lambda(1-\alpha)$. Thus to solve the requested minimization problem given $\mathbf{y}$, $\mathbf{X}$, λ and α perform the following steps

- Augment $\mathbf{y}$ with p additional zeros to get $\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}$.
- Augment $\mathbf{X}$ with the multiple of the $p \times p$ identity matrix $\sqrt{\lambda\alpha}\mathbf{I}$ to get $\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \gamma\mathbf{I} \end{bmatrix}$.
- Set $\tilde{\lambda} = \lambda(1-\alpha)$.
- Solve the LASSO minimization problem with input $\tilde{\mathbf{y}}$, $\tilde{\mathbf{X}}$ and $\tilde{\lambda}$.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

Optimization and Machine Learning, Spring 2020
Homework 2
(Due Wednesday, Apr. 1 at 11:59pm (CST))

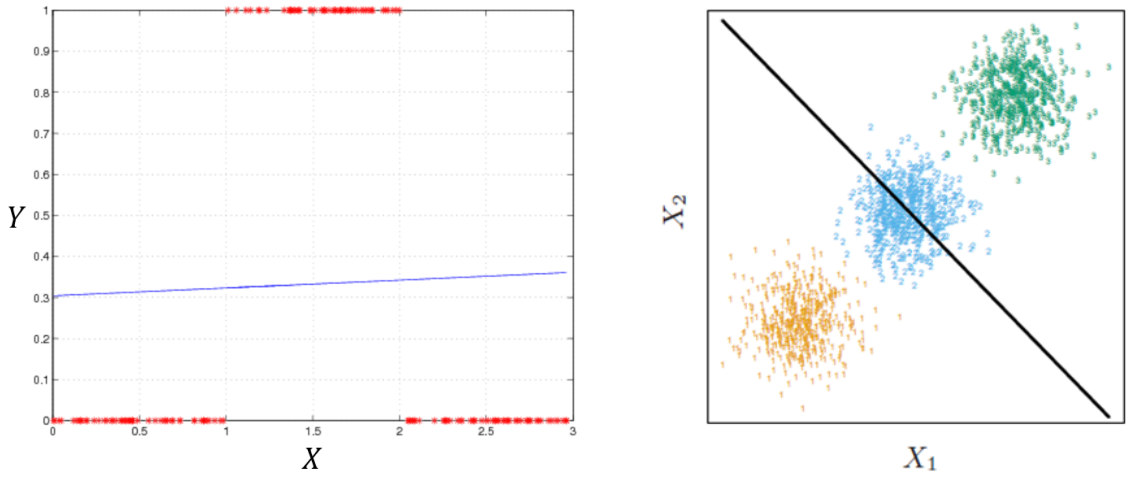
April 5, 2020

1. Suppose that we have N training samples, in which each sample is composed of p input variable and one categorical response with K states.

   (a) Please define this multi-class classification problem, and solve it by ridge regression. (4 points)
   Input:
   $$\mathbf{X} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix},$$
   where x_i is the i-th observation with p parameter.
   Output:
   $$\mathbf{Y} = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix},$$
   where $y_i = (0, \cdots, 0, 1, 0, \cdots, 0)^T$, with its k-th element being 1, indicating that the k-th class is associated with the i-th observation x_i.
   By minimizing the following objective function,
   $$||\mathbf{XB} - \mathbf{Y}||_F^2 + \lambda||\mathbf{B}||_F^2,$$
   we can get the solution $\mathbf{B} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{X}^T\mathbf{Y}$, where $\lambda > 0$ denotes the regularization parameter.

   (b) Please make the prediction of a testing sample $x \in \mathbb{R}^p$ based on your model in (a). (3 points)
   The prediction is made by
   $$\hat{y} = \arg\max_k \hat{f}_k(x),$$
   where $\hat{f}_k(x)$ is the k-th element of
   $$\hat{f}(x) = x^T\mathbf{B} = \begin{pmatrix} \hat{f}_1(x) \\ \hat{f}_2(x) \\ \vdots \\ \hat{f}_K(x) \end{pmatrix}.$$

   (c) Is there any limitation on your model? If yes, please explain the problem by drawing a picture. (3 points)
   The masking problem:



(a) Two classes (b) Three classes

Figure 1: Illustion of the masking problem in linear regression for classification.

(d) Can you propose a model to overcome this limitation? If yes, please derive the decision boundary between an arbitrary class-pair. (5 points)
Linear discriminant analysis (LDA). In LDA, the decision boundary between two arbitrary classes A and B is

$$\mathbf{x}^T\hat{\Sigma}^{-1}(\hat{\mu}_A - \hat{\mu}_B) + \left(\ln(\frac{\Pr(A)}{\Pr(B)}) - \frac{\hat{\mu}_A\hat{\Sigma}^{-1}\hat{\mu}_A - \hat{\mu}_B\hat{\Sigma}^{-1}\hat{\mu}_B}{2}\right) = 0.$$

(e) Can you revise your model in (d) by strength or weaken its assumptions? If yes, please tell the difference between your models in (d) and (e). (5 points)
We can use quadratic discriminative analysis (QDA) for classification.
Difference:

- LDA and QDA both assume that the class conditional probability distributions are normally distributed with different means μ_k, but LDA is different from QDA in that it requires all of the distributions to share the same covariance matrix Σ and QDA requires all of the distribution to have different covariance matrix Σ_k.
- The decision boundary is linear in LDA and quadratic in QDA.
- The number of estimated parameters is $p \times (K + p)$ in LDA and $K \times p \times (p + 1)$ in QDA.

2. Given an random variable, we have N i.i.d. observations by repeated experiments.

(a) If the variable is boolean, please calculate the log-likelihood function. (4 points)
Let X be a boolean random variable which can take either value 1 or 0, and let $\theta = \Pr(X = 1)$ refer to the true. Given an i.i.d dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, we observe $X = 1$ in a total of α_1 times, and $X = 0$ in a total of α_0 times. We denote the likelihood function by $L(\theta) = \Pr(\mathcal{D}|\theta)$:

$$L(\theta) = \Pr(\mathcal{D}|\theta) = \prod_{i=1}^{N} \Pr(X = x_i|\theta) = \theta^{\alpha_1}(1-\theta)^{\alpha_0}, \quad x_i \in \{0, 1\}.$$

By taking log on both sides, we have the log-likelihood function, i.e.,

$$\ell(\theta) = \ln L(\theta) = \alpha_1 \ln\theta + \alpha_0 \ln(1-\theta).$$

(b) If the variable is categorical, please calculate the log-likelihood function. (4 points)
Suppose that $X \in \{1, 2, ..., K\}$, and $\theta = \{\theta_1, ..., \theta_K\}$, in which the k-th element $\theta_k = \Pr(X = k)$. The likelihood function $L(\theta) = \Pr(\mathcal{D}|\theta)$ is then derived in the similar way with (a),

$$L(\theta) = \Pr(\mathcal{D}|\theta) = \prod_{i=1}^{N} \Pr(X = x_i|\theta) = \prod_{k=1}^{K} \theta_k^{\alpha_k}, \quad x_i \in \{1, ..., K\}.$$

where α_k counts the number of $x_i = k$ in $\mathcal{D}$, $\forall i, k$. Thus, the log-likelihood function is calculated by

$$\ell(\theta) = \ln L(\theta) = \sum_{k=1}^{K} \alpha_k \ln \theta_k.$$

(c) If the variable is continuous and follows Gaussian distribution, please calculate the log-likelihood function. (5 points)
Let X be a Gaussian random variable parameterized by mean μ and variance σ. Its PDF is given by

$$\mathcal{N}\left(X|\mu, \sigma^2\right) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}.$$

Under the i.i.d. assumption, the likelihood function $L(\mu, \sigma)$ is expressed as follows,

$$L(\mu, \sigma) = \Pr(\mathcal{D}|\mu, \sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_i-\mu)^2},$$

and the log-likelihood function becomes

$$\ell(\mu, \sigma) = \ln L(\mu, \sigma) = \frac{N}{2} \ln \frac{1}{2\pi\sigma^2} - \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{2\sigma^2}.$$

(d) Please discuss the difference between Maximum Likelihood Estimation (MLE) and Maximum a Posterior (MAP) estimation based on ONE of your results in (a), (b) and (c). (7 points)
MLE seeks an estimation of θ that maximizes the conditional probability $\Pr(\mathcal{D}|\theta)$; in contrast, MAP aims to estimate θ by maximizing its posterior $\Pr(\theta|\mathcal{D})$, leading to $\Pr(\theta|\mathcal{D}) \propto \Pr(\mathcal{D}|\theta)\Pr(\theta)$.
We can also see the difference by analyzing the results of (a), in which MLE produces

$$\hat{\theta}^{MLE} = \frac{\alpha_1}{\alpha_1 + \alpha_0},$$

while MAP gives rise to

$$\hat{\theta}^{MAP} = \frac{\alpha_1 + \beta_1 - 1}{(\alpha_1 + \beta_1 - 1) + (\alpha_0 + \beta_0 - 1)},$$

with the Beta prior $Beta(\beta_1, \beta_0)$.

3. Given the input variables $X \in \mathbb{R}^p$ and a response variable $Y \in \{0, 1\}$, the Expected Prediction Error (EPE) is defined by

$$\text{EPE} = \mathbb{E}[L(Y, \hat{Y}(X))],$$

where $\mathbb{E}(\cdot)$ denotes the expectation over the joint distribution $\Pr(X, Y)$, and $L(Y, \hat{Y}(X))$ is a loss function measuring the difference between the estimated $\hat{Y}(X)$ and observed Y.

(a) Given the zero-one loss

$$L(k, \ell) = \begin{cases} 1 & \text{if } k \neq \ell \\ 0 & \text{if } k = \ell, \end{cases}$$

please derive the Bayes classifier $\hat{Y}(x) = \text{argmax}_{k \in \{0,1\}} \Pr(Y = k|X = x)$ by minimizing EPE. (2 points)
Without loss of generality, we consider $Y \in \{1, 2, ..., M\}$, and rewrite EPE as follows

$$\begin{aligned} \text{EPE} &= \mathbb{E}[L(Y, \hat{Y}(X)] \\ &= \int_x \left[\sum_{m=1}^{M} L\left(Y = m, \hat{Y}(x)\right) \Pr(Y = m|X = x) \right] dx \\ &= \int_x \left[1 - \Pr\left(Y = \hat{Y}(x)|X = x\right) \right] dx. \end{aligned}$$

Therefore,

$$\hat{Y}(x) = \text{argmin EPE} = \text{argmax}_{m \in \{1,...,M\}} \Pr(Y = m|X = x).$$

(b) Please define a function which enables to map the range of an arbitrary linear function to the range of a probability. (2 points)
Given an arbitrary liner function,

$$f(X) = \beta_0 + X^\top \beta \in (-\infty, +\infty),$$

the required function can be defined by

$$\Pr(Y|X) = \frac{\exp(f(X))}{1 + \exp(f(X))} \in (0, 1).$$

(c) Based on the function you defined in (b), please approximate the Bayes classifier in (a) by a linear function between X and Y, and derive its decision boundary. (4 points)
Based on (a), we have

$$\Pr(Y = 0|X) = \frac{\exp(f(X))}{1 + \exp(f(X))},$$
$$\Pr(Y = 1|X) = 1 - \Pr(Y = 0|X) = \frac{1}{1 + \exp(f(X))}.$$

Thus, using the Bayes classifier in (a), we assign the label $Y = 0$ if the following conditions hold:

$$\begin{aligned} & 1 < \frac{P(Y = 0|\boldsymbol{X})}{P(Y = 1|\boldsymbol{X})} \\ \Longrightarrow\ & 0 < \ln \exp(f(\boldsymbol{x})) \\ \Longrightarrow\ & 0 < f(\boldsymbol{x}), \end{aligned}$$

and assign $Y = 1$ otherwise. Hence, we obtain the linear decision boundary $\{X|\beta_0 + X^\top \beta = 0\}$.

(d) If each element of X is boolean, please show how many independent parameters are needed in order to estimate $\Pr(Y|X)$ directly; and is there any way to reduce its number? If yes, please describe your way mathematically. (4 points)
Given $X \in \{0, 1\}^p$ and $Y \in \{0, 1\}$, we need 2^p parameters to estimate $\Pr(Y = 1|X)$, and another 2^p parameters to estimate $\Pr(Y = 0|X)$. However, because of $\Pr(Y = 0|X) = 1 - \Pr(Y = 1|X)$, there are 2^p independent parameters in total.
To reduce the number of parameters, conditional independent assumption is applied, such that

$$\begin{aligned} \Pr(Y|X) &\propto \Pr(X_1, \ldots, X_p|Y)\Pr(Y) \\ &= \prod_{j=1}^{p} \Pr(X_j|Y)\Pr(Y), \end{aligned}$$

according to which we only need to estimate $2p$ independent parameters.

(e) Based on your results in (d) and the Bayes theorem, please develop a classifier with a linear number of parameters w.r.t. p, and estimate these parameters by MLE. (5 points)
Naive Bayes:
Based on the results in (d) and the Bayes theorem, we immediately obtain the naive Bayes classifier:

$$\hat{y} = \text{argmax}_{m \in \{0,1\}} \Pr(Y = m) \prod_{j=1}^{p} \Pr\left(X_j = k|Y = m\right), \quad k \in \{0, 1\}$$

MLE:
According to our discussion in Q2 (a) and (d), $\Pr(X_j = k|Y = m)$ $(k, m \in \{0, 1\})$ is estimated on a training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ by

$$\widehat{\Pr}\left(X_j = k|Y = m\right) = \frac{\sum_{i=1}^N \mathbf{1}_{x_{ij}=k} \mathbf{1}_{y_i=m}}{\sum_{i=1}^N \mathbf{1}_{y_i=m}},$$
$$\widehat{\Pr}\left(Y = m\right) = \frac{\sum_{i=1}^N \mathbf{1}_{y_i=m}}{N},$$

where $\mathbf{1}_{(\cdot)}$ denotes the indicator function.

(f) Please find at least three different points between your developed models in (c) and (e). (3 points)
Difference:

- Naive Bayes in (e) assumes that the random variables are conditional independent given Y, whereas logistic regression in (c) does not hold such assumption.
- Naive Bayes in (e) estimates the parameters of $\Pr(X|Y)$ and $\Pr(Y)$, whereas logistic regression in (c) choose to directly approximate $\Pr(Y|X)$ by a linear function.
- Naive Bayes is a generative model since it models $\Pr(X, Y)$, while logistic regression is a discriminative model as it approximates $\Pr(Y|X)$.
- Two models will converge toward their asymptotic accuracies at different rates.

4. Consider 12 labeled data points sampled from three distinct classes:

$$\text{Class 0}: \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}-2\\0\end{bmatrix}, \begin{bmatrix}5\\3\end{bmatrix}, \begin{bmatrix}-3\\-5\end{bmatrix} \qquad \text{Class 1}: \begin{bmatrix}\sqrt{2}\\\sqrt{2}\end{bmatrix}, \begin{bmatrix}-\sqrt{2}\\\sqrt{2}\end{bmatrix}, \begin{bmatrix}4\sqrt{2}\\-\sqrt{2}\end{bmatrix}, \begin{bmatrix}-4\sqrt{2}\\-\sqrt{2}\end{bmatrix}, \qquad \text{Class 2}: \begin{bmatrix}3\\5\end{bmatrix}, \begin{bmatrix}-1\\3\end{bmatrix}, \begin{bmatrix}8\\6\end{bmatrix}, \begin{bmatrix}0\\-2\end{bmatrix}$$

(a) For each class $C \in [0, 1, 2]$, compute the class sample mean μ_C, the class sample covariance matrix Σ_C, and the estimate of the prior probability π_C that a point belongs to class C. (6 points)
Solution:

$$\text{Class 0}: \mu_0 = \begin{bmatrix}0\\0\end{bmatrix}, \Sigma_0 = \begin{bmatrix}\frac{38}{3} & 10\\10 & \frac{38}{3}\end{bmatrix}, \pi_0 = \frac{1}{3},$$
$$\text{Class 1}: \mu_1 = \begin{bmatrix}0\\0\end{bmatrix}, \Sigma_1 = \begin{bmatrix}\frac{68}{3} & 0\\0 & \frac{8}{3}\end{bmatrix}, \pi_1 = \frac{1}{3},$$
$$\text{Class 2}: \mu_2 = \begin{bmatrix}2.5\\3\end{bmatrix}, \Sigma_2 = \begin{bmatrix}\frac{49}{3} & 10\\10 & \frac{38}{3}\end{bmatrix}, \pi_1 = \frac{1}{3}.$$

(b) Suppose that we apply LDA to classify the data given in part (a). Will this get the good decision boundary? Briefly explain your answer. (4 points)
The discriminant functions for classes 0 and 1 would have the exact same mean, so there would be no decision boundary between them.

5. We have two classes, named N for normal and E for exponential. For the former class ($Y = N$), the prior probability is $\pi_N = P(Y = N) = \frac{\sqrt{2\pi}}{1+\sqrt{2\pi}}$ and the class conditional $P(X|Y = N)$ has the normal distribution $N(0, \sigma^2)$. For the latter, the prior probability is $\pi_E = P(Y = E) = \frac{1}{1+\sqrt{2\pi}}$ and the class conditional has the exponential distribution.

$$P(X = x|Y = E) = \begin{cases}\lambda e^{-\lambda x} & \text{if } x \geq 0\\ 0 & \text{if } x < 0\end{cases}$$

Write an equation in x for the decision boundary. (Only the positive solutions of your equation will be relevant; ignore all $x < 0$.) Simplify the equation until it is quadratic in x. (You dont need to solve the quadratic equation. It should contain the constants σ and λ. Ignore the fact that 0 might or might not also be a point in the decision boundary.) (10 points)
Solution:

$$P(Y = N|X = x) = P(Y = E|X = x)$$
$$P(X = x|Y = N)P(Y = N) = P(X = x|Y = E)P(Y = E)$$
$$\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{x^2}{2\sigma^2})\frac{\sqrt{2\pi}}{1+\sqrt{2\pi}} = \lambda e^{-\lambda x}\frac{1}{1+\sqrt{2\pi}}$$
$$-\ln\sigma - \frac{x^2}{2\sigma^2} = \ln\lambda - \lambda x$$
$$\frac{x^2}{2\sigma^2} - \lambda x + \ln\sigma + \ln\lambda = 0.$$

6. Given data $\{(x_i, y_i) \in R^d \times \{0, 1\}\}_{i=1}^n$ and a query point x, we choose a parameter vector θ to minimize the loss (which is simply the negative log likelihood, weighted appropriately):

$$l(\theta; x) = -\sum_{i=1}^{n} w_i(x)[y_i \log(\mu(x_i)) + (1 - y_i)\log(1 - \mu(x_i))]$$

where

$$\mu(x_i) = \frac{1}{1+e^{-\theta \cdot x_i}}, w_i(x) = \exp(-\frac{||x-x_i||^2}{2\tau})$$

where τ s a hyperparameter that must be tuned. Note that whenever we receive a new query point x, we must solve the entire problem again with these new weights $w_i(x)$.

(a) Given a data point x, derive the gradient of $l(\theta; x)$ with respect to θ. (4 points)

$$\begin{aligned}\nabla_\theta l(\theta; x) &= -\sum_{i=1}^{n} w_i(x)(y_i - \mu(x_i))x_i \\ &= -\mathbf{X}^T z,\end{aligned}$$

where $z_i = w_i(x)(y_i - \mu(x_i))$.

(b) Given a data point x, derive the Hessian of $l(\theta; x)$ with respect to θ. (4 points)

$$\begin{aligned}H_\theta l(\theta; x) &= -\sum_{i=1}^{n} w_i(x)\mu(x_i)(1-\mu(x_i))x_i x_i^T \\ &= \mathbf{X}^T D\mathbf{X},\end{aligned}$$

where $D_{ii} = w_i(x)\mu(x_i)(1-\mu(x_i))$, $D_{ij} = 0$ if $i \neq j$.

(c) Given a data point x, write the update formula for Newtons method. (2 points)

$$\theta^{(t+1)} = \theta^{(t)} + [\mathbf{X}^T D\mathbf{X}]^{-1}\mathbf{X}^T z.$$

7. Now we discuss Bayesian inference in coin flipping. Let's denote the number of heads and the total number of trials by N_1 and N, respectively.

(a) Please derive the MAP estimation based on the prior $p(\theta) = \text{Beta}(\theta|\alpha, \beta)$. (4 points)
The Beta distribution is defined by

$$P(\theta) = \text{Beta}(\alpha, \beta) = \frac{\theta^{\beta-1}(1-\theta)^{\alpha-1}}{B(\alpha, \beta)}.$$

Combining with the expression for $P(D|\theta)$, we have:

$$\begin{aligned}\hat{\theta}^{MAP} &= \arg\max_\theta P(D|\theta)P(\theta) \\ &= \arg\max_\theta \theta^{N_1}(1-\theta)^{N_0}\frac{\theta^{\beta-1}(1-\theta)^{\alpha-1}}{B(\alpha, \beta)} \\ &= \arg\max_\theta \frac{\theta^{N_1+\beta-1}(1-\theta)^{N_0+\alpha-1}}{B(\alpha, \beta)} \\ &= \arg\max_\theta \theta^{N_1+\beta-1}(1-\theta)^{N_0+\alpha-1}.\end{aligned}$$

We calculate the derivative of the log of the likelihood function:

$$\begin{aligned}\frac{\partial \ell(\theta)}{\partial \theta} &= \frac{\partial \ln P(D|\theta)P(\theta)}{\partial \theta} \\ &= \frac{\partial \ln\left[\theta^{N_1+\beta-1}(1-\theta)^{N_0+\alpha-1}\right]}{\partial \theta} \\ &= \frac{\partial\left[N_1 \ln\theta + N_0 \ln(1-\theta)\right]}{\partial \theta} \\ &= (N_1+\beta_1-1)\frac{\partial \ln\theta}{\partial \theta} + (N_0+\alpha-1)\frac{\partial \ln(1-\theta)}{\partial \theta} \\ &= (N_1+\beta_1-1)\frac{\partial \ln\theta}{\partial \theta} + (N_0+\alpha-1)\frac{\partial \ln(1-\theta)}{\partial(1-\theta)}\cdot\frac{\partial(1-\theta)}{\partial \theta} \\ \frac{\partial \ell(\theta)}{\partial \theta} &= (N_1+\beta_1-1)\frac{1}{\theta} - (N_0+\alpha-1)\frac{1}{(1-\theta)}.\end{aligned}$$

Therefore,

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(D|\theta)P(\theta) = \frac{N_1 + \beta - 1}{N + \beta + \alpha - 2}.$$

(b) Please derive the MAP estimation based on the following prior:

$$p(\theta) = \begin{cases} 0.5 & \text{if } \theta = 0.5 \\ 0.5 & \text{if } \theta = 0.4 \\ 0 & \text{otherwise,} \end{cases}$$

that believes the coin is fair, or is slightly biased towards tails. (4 points)
With the prior, the posterior becomes

$$P(D|\theta)P(\theta) = \begin{cases} 0.5 \cdot 0.5^{N_1}(1-0.5)^{N_0} & \theta = 0.5 \\ 0.5 \cdot 0.4^{N_1}(1-0.4)^{N_0} & \theta = 0.4 \\ 0 & \text{otherwise} \end{cases} = \begin{cases} 0.5^{N+1} & \theta = 0.5 \\ 0.5 \cdot 0.4^{N_1}0.6^{N-N_1} & \theta = 0.4 \\ 0 & \text{otherwise} \end{cases}$$

Since the value of θ only can be taken 0.5 or 0.4, we just need to compare two posteriors as follows:

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(D|\theta)P(\theta) = \begin{cases} 0.5 & \text{if } 0.5^{N+1} > 0.5 \cdot 0.4^{N_1}0.6^{N-N_1}, \\ 0.4 & \text{if } 0.5^{N+1} < 0.5 \cdot 0.4^{N_1}0.6^{N-N_1}. \end{cases}$$

Here, we don't consider the case of $0.5^{N+1} = 0.5 \cdot 0.4^{N_1}0.6^{N-N_1}$. After some simple computations, we have the solution:

$$\hat{\theta}^{MAP} = \begin{cases} 0.5 & \text{if } N < \frac{\ln 3 - \ln 2}{\ln 6 - \ln 5} N_1, \\ 0.4 & \text{if } N > \frac{\ln 3 - \ln 2}{\ln 6 - \ln 5} N_1. \end{cases}$$

(c) Suppose the true parameter is $\theta = 0.41$. Which prior leads to a better estimate when N is small? Which prior leads to a better estimate when N is large? (2 points)
When N is small, the prior in (b) leads a better estimate since the prior is a summary of our subjective beliefs about the data. When N is large, the estimate in (a) is better according to the law of large number.

Optimization and Machine Learning, Spring 2020
Reference Solutions for Homework 3

1. (a) Consider the linear regression from a probabilistic perspective. Suppose we are given a set of N observations of the input vector $\mathbf{x}$, which we denote collectively by a data matrix $\mathbf{X}$ whose n-th row is $\mathbf{x}_n^T$ with $n = 1, \cdots, N$. The corresponding target values are $\boldsymbol{t} = (t_1, \cdots, t_N)^T$. We can express uncertainty over the value of target variable using a probability distribution. Assume that given the data $\mathbf{x}_n$ and coefficient vector $\mathbf{w}$, the corresponding value of t_n has a Gaussian distribution with variance σ^2. If the data are assumed to be drawn independently, then the likelihood function is given by

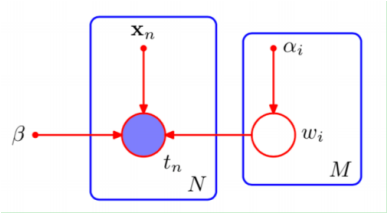
$$p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} p(t_n \mid \mathbf{x}_n, \mathbf{w}, \sigma^2). \tag{1}$$

Next we similarly introduce a prior distribution over the parameter vector $\mathbf{w}$, we shall consider a zero-mean Gaussian prior with variance α_i for each w_i. Assume that the parameter variables are independent. Thus the parameter prior takes the form

$$p(\mathbf{w} \mid \alpha) = \prod_{n=1}^{M} \mathcal{N}(w_i \mid 0, \alpha_i^{-1}). \tag{2}$$

Draw a directed probabilistic graphical model corresponding to the relevance vector machine described by equations (1) and (2). (5 points)

Solution: Introduce a graphical notation that allows such multiple nodes to be expressed more compactly, in which we draw a single representative node tn and then surround this with a box, called a plate, labelled with N indicating that there are N nodes of this kind.



(b) Consider the model defined in (a). Suppose we are given a new input data $\hat{x}$ and we wish to find the corresponding probability distribution for $\hat{t}$ conditioned on the observed data. The graphical model that describes this problem is shown in following Fig. 1. Please give the corresponding joint distribution of all of the random variables in this model and conditioned on the deterministic parameters, i.e., $p(\hat{t}, \mathbf{t}, \mathbf{w} \mid \hat{x}, \mathbf{x}, \alpha, \sigma^2)$. (5 points)

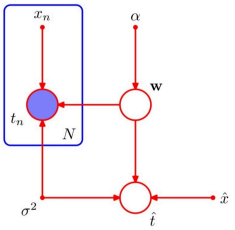


Figure 1: The graphical model.

Solution:

$$p(\hat{t}, \mathbf{t}, \mathbf{w} \mid \hat{x}, \mathbf{x}, \alpha, \sigma^2) = \left[\prod_{n=1}^{N} p(t_n \mid \mathbf{x}_n, \mathbf{w}, \sigma^2)\right] p(\mathbf{w} \mid \alpha) p(\hat{t} \mid \hat{x}, \mathbf{w}, \sigma^2).$$

2. According to the following Fig. 2, use the D-separation to analyze the following cases:

   (a) Given x_4, $\{x_1, x_2\}$ and $\{x_6, x_7\}$ are conditionally independent. (5 points)

   (b) Given $\{x_6, x_7\}$, x_3 and x_5 are conditionally independent. (5 points)

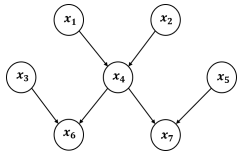


Figure 2: The Bayesian network for questions 2 and 3.

Solution:

(a) The statement is True. According to D-separation, $\{x_1, x_2\}$ and $\{x_6, x_7\}$ can be regarded as two sets A and B. All the arrows on the path form A to B meet head-to-tail, therefore all the paths are blocked given x_4.

(b) The statement is False. The arrow on the path from x_3 to x_4 meets head-to-head. Since the node x_6 is observed, the path from x_3 to x_4 is not blocked. The path from x_3 to x_4 is the same. The path from x_6 to x_7 is also unblocked, therefore x_3 and x_5 are not conditionally independent.

3. According to the Fig. 2, if all the nodes are observed and boolean variables, please complete the process of learning the parameter $\theta_{x_4|i,j}$ by using **MLE**, where $\theta_{x_4|i,j} = p(x_4 = 1 \mid x_1 = i, x_2 = j), i, j \in \{0, 1\}$. (15 points)

   Solution: Suppose we observed K data points. Let $\theta = \{\theta_{x_1}, \theta_{x_2}, \theta_{x_3}, \theta_{x_5}, \theta_{x_4|i,j}, \theta_{x_6|i,j}, \theta_{x_7|i,j}\}$, then

$$\begin{aligned}
\log p(\mathcal{D} \mid \theta) &= \log \prod_{k=1}^{K} p(x_{1k}, x_{2k}, x_{3k}, x_{4k}, x_{5k}, x_{6k}, x_{7k} \mid \theta) \\
&= \log \prod_{k=1}^{K} p(x_{1k} \mid \theta) p(x_{2k} \mid \theta) p(x_{3k} \mid \theta) p(x_{5k} \mid \theta) p(x_{4k} \mid x_{1k}, x_{2k}, \theta) p(x_{6k} \mid x_{3k}, x_{4k}, \theta) p(x_{7k} \mid x_{4k}, x_{5k}, \theta) \\
&= \sum_{k=1}^{K} \log p(x_{1k} \mid \theta) + \log p(x_{2k} \mid \theta) + \log p(x_{3k} \mid \theta) + \log p(x_{5k} \mid \theta) + \log p(x_{4k} \mid x_{1k}, x_{2k}, \theta) \\
&\quad + \log p(x_{6k} \mid x_{3k}, x_{4k}, \theta) + \log p(x_{7k} \mid x_{4k}, x_{5k}, \theta).
\end{aligned}$$

   Then we derive the gradient of $\log p(\mathcal{D} \mid \theta)$ with respect to $\theta_{x_4|i,j}$

$$\frac{\partial \log p(\mathcal{D} \mid \theta)}{\partial \theta_{x_4|i,j}} = \sum_{ik=1}^{K} \frac{\partial p(x_{4k} \mid x_{1k}, x_{2k}, \theta)}{\partial \theta_{x_4|i,j}}$$

   Set the derivative to 0 and then obtain the parameter $\theta_{x_4|i,j}$

$$\theta_{x_4|i,j} = \frac{\sum_{k=1}^{K} \mathbb{I}(x_{4k} = 1, x_{1k} = i, x_{2k} = j)}{\sum_{k=1}^{K} \mathbb{I}(x_{1k} = i, x_{2k} = j)},$$

   where $\mathbb{I}(\cdot)$ is the indicator function.

4. Define a Bayesian network with five discrete variables, represented by $\{F, A, S, H, N\}$. $\{F, A, H, N\}$ are 0/1 binary variables and $S \in \{0, 1, 2\}$, as illustrated in Fig. 3. Among them, $\{F, A, N\}$ are observed variables and $\{S, H\}$ are latent variables. Now we implement EM algorithm for this model.

   (a) If all five variables are observed, derive MLE of this model. You should state the close-form solution for each parameter you define. (5 points)

   (b) At least how many parameters should be defined for EM algorithm? (2 points)

   (c) Derive the E-step. You should enumerate each term. (4 points)

   (d) Derive the M-step. (4 points)

   Solution:

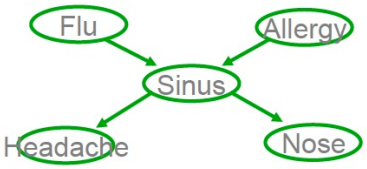


Figure 3: The Bayesian network for question 4.

(a) Define $\theta_f = P(F=1), \theta_a = P(A=1), \theta_{s|f,a} = P(S=s|F=f, A=a), \theta_{h|s} = P(H=1|S=s), \theta_{n|s} = P(N=1|S=s)$. Suppose there are K data points. The likelihood function is

$$\begin{aligned} l(\theta) &= \prod_{k=1}^{K} P(x_k|\theta) = \prod_{k=1}^{K} P(f_k, a_k, s_k, h_k, n_k|\theta) \\ &= \prod_{k=1}^{K} P(f_k)P(a_k)P(s_k|f_k, a_k)P(h_k|s_k)P(n_k|s_k) \end{aligned}$$

Set the derivative of log likelihood function with respect to each parameter to 0, the solutions are:

$$\theta_f = \frac{\sum_{k=1}^{K} \delta(f_k = 1)}{K},$$

$$\theta_a = \frac{\sum_{k=1}^{K} \delta(a_k = 1)}{K},$$

$$\theta_{s|f,a} = \frac{\sum_{k=1}^{K} \delta(s_k = s|f_k = f, a_k = a)}{\sum_{k=1}^{K} \delta(f_k = f, a_k = a)},$$

$$\theta_{h|s} = \frac{\sum_{k=1}^{K} \delta(h_k = 1|s_k = s)}{\sum_{k=1}^{K} \delta(s_k = s)},$$

$$\theta_{n|s} = \frac{\sum_{k=1}^{K} \delta(n_k = 1|s_k = s)}{\sum_{k=1}^{K} \delta(s_k = s)}.$$

(b) At least 16 variables for θ.

(c) In E-step, calculate $P(S, H|F, A, N, \theta)$.

$$P(s_k = 0, h_k = 0|f_k, a_k, n_k, \theta) = \frac{P(s_k = 0, h_k = 0, f_k, a_k, n_k|\theta)}{\sum_{i=0}^{2}\sum_{j=0}^{1} P(s_k = i, h_k = j, f_k, a_k, n_k|\theta)},$$

$$P(s_k = 0, h_k = 1|f_k, a_k, n_k, \theta) = \frac{P(s_k = 0, h_k = 1, f_k, a_k, n_k|\theta)}{\sum_{i=0}^{2}\sum_{j=0}^{1} P(s_k = i, h_k = j, f_k, a_k, n_k|\theta)},$$

$$P(s_k = 1, h_k = 0|f_k, a_k, n_k, \theta) = \frac{P(s_k = 1, h_k = 0, f_k, a_k, n_k|\theta)}{\sum_{i=0}^{2}\sum_{j=0}^{1} P(s_k = i, h_k = j, f_k, a_k, n_k|\theta)},$$

$$P(s_k = 1, h_k = 1|f_k, a_k, n_k, \theta) = \frac{P(s_k = 1, h_k = 1, f_k, a_k, n_k|\theta)}{\sum_{i=0}^{2}\sum_{j=0}^{1} P(s_k = i, h_k = j, f_k, a_k, n_k|\theta)},$$

$$P(s_k = 2, h_k = 0|f_k, a_k, n_k, \theta) = \frac{P(s_k = 2, h_k = 0, f_k, a_k, n_k|\theta)}{\sum_{i=0}^{2}\sum_{j=0}^{1} P(s_k = i, h_k = j, f_k, a_k, n_k|\theta)},$$

$$P(s_k = 2, h_k = 1|f_k, a_k, n_k, \theta) = \frac{P(s_k = 2, h_k = 1, f_k, a_k, n_k|\theta)}{\sum_{i=0}^{2}\sum_{j=0}^{1} P(s_k = i, h_k = j, f_k, a_k, n_k|\theta)}.$$

(d) In M-step, choose θ' which maximize $E_{P(S,H|F,A,N,\theta)} \log P(S, H, F, A, N|\theta')$, where

$$\begin{aligned} &E_{P(S,H|F,A,N,\theta)} \log P(S, H, F, A, N|\theta') \\ &= \sum_{k=1}^{K}\sum_{i=0}^{2}\sum_{j=0}^{1} P(s_k = i, h_k = j|f_k, a_k, n_k, \theta)[\log P(f_k) + \log P(a_k) + \log P(s_k|f_k, a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)]. \end{aligned}$$

5. Consider a set of K binary variables x_i, where $i = \{1, ..., K\}$, each variable $x_i \sim Bern(\mu_i)$. So $P(\mathbf{x}|\boldsymbol{\mu}) = \prod_{i=1}^{K} \mu_i^{x_i}(1-\mu_i)^{1-x_i}$, where $\mathbf{x} = (x_1, ..., x_K)^T$ *and* $\boldsymbol{\mu} = (\mu_1, ..., \mu_K)^T$. The mean and covariance of this distribution are easily seen to be $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$ and $\text{cov}[\mathbf{x}] = \text{diag}\{\mu_i(1-\mu_i)\}$.

   Now define a finite mixture of N Bernoullis given by $P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \pi_n P(\mathbf{x}|\boldsymbol{\mu_n})$ where $\boldsymbol{\mu} = \{\boldsymbol{\mu_1}, ..., \boldsymbol{\mu_N}\}$, $\boldsymbol{\pi} = \{\pi_i, ..., \pi_N\}$ and $P(\mathbf{x}|\boldsymbol{\mu_n}) = \prod_{i=1}^{K} \mu_{ni}^{x_i}(1-\mu_{ni})^{1-x_i}$.

   (a) Derive the mean of the mixture distribution. (5 points)

   (b) Show the covariance of the mixture distribution equals $\sum_{n=1}^{N} \pi_n\{\boldsymbol{\Sigma_n} + \boldsymbol{\mu_n}\boldsymbol{\mu_n}^T\} - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T$, where $\boldsymbol{\Sigma_n} = \text{diag}\{\mu_{ni}(1-\mu_{ni})$. (5 points)

   Solution:

   (a)

$$\begin{aligned}\mathbb{E}(P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\pi})) &= \sum_{n=1}^{N} \pi_n \mathbb{E}(P(\mathbf{x}|\boldsymbol{\mu_n})) \\ &= \sum_{n=1}^{N} \pi_n \boldsymbol{\mu_n}\end{aligned}$$

   (b)

$$\begin{aligned}\text{cov}[\mathbf{x}] &= \mathbb{E}[\mathbf{x}\mathbf{x}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T \\ &= \sum_{n=1}^{N} \pi_n \mathbb{E}_n[\mathbf{x}\mathbf{x}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T \\ &= \sum_{n=1}^{N} \pi_n\{\boldsymbol{\Sigma_n} + \boldsymbol{\mu_n}\boldsymbol{\mu_n}^T\} - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T\end{aligned}$$

6. Derive EM algorithm for the mixture of Bernoulli distributions above. There are D data points in total, where $\mathbf{X} = \{\mathbf{x_1}, ..., \mathbf{x_D}\}$. (15 points)

   Solution: The log likelihood function for the model is

$$\log P(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{d=1}^{D} \log\{\sum_{n=1}^{N} \pi_n P(\mathbf{x_d}|\boldsymbol{\mu_n})\}.$$

   Assume the latent variable $\mathbf{z} = (z_1, ..., z_N)^T$ is a binary N-dimensional variable having only a single component equal to 1. So we have

$$P(\mathbf{x}|\mathbf{z}, \boldsymbol{\mu}) = \prod_{n=1}^{N} P\left(\mathbf{x}|\boldsymbol{\mu}_n\right)^{z_n}, \;\; P(\mathbf{z}|\boldsymbol{\pi}) = \prod_{n=1}^{N} \pi_n^{z_n}.$$

   If we form the product of $P(\mathbf{x}|\mathbf{z}, \boldsymbol{\mu})$ and $P(\mathbf{z}|\boldsymbol{\pi})$ and then marginalize over $\mathbf{z}$, then we obtain $P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \pi_n P\left(\mathbf{x}|\boldsymbol{\mu}_n\right)$.
   Then the log likelihood function for complete-data is

$$\log P(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{d=1}^{D}\sum_{n=1}^{N} z_{dn}\{\log \pi_n + \sum_{k=1}^{K}[x_{dk} \log \mu_{nk} + (1-x_{dk}) \log(1-\mu_{nk})]\}.$$

   Expecting over $\mathbf{Z}$, we have

$$\mathbb{E}_{\mathbf{Z}}[\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\pi})] = \sum_{d=1}^{D}\sum_{n=1}^{N} \gamma\left(z_{dn}\right)\left\{\log \pi_n + \sum_{k=1}^{K}\left[x_{dk} \log \mu_{nk} + (1-x_{dk}) \log\left(1-\mu_{nk}\right)\right]\right\},$$

   where $\gamma\left(z_{dn}\right) = \mathbb{E}[z_{dn}] = \frac{\sum_{z_{dn}} z_{dn}[\pi_n p(\mathbf{x}_d|\boldsymbol{\mu}_n)]^{z_{dn}}}{\sum_{z_{dj}}[\pi_j p(\mathbf{x}_d|\boldsymbol{\mu}_j)]^{z_{dj}}} = \frac{\pi_n p(\mathbf{x}_d|\boldsymbol{\mu}_n)}{\sum_{j=1}^{N} \pi_j p(\mathbf{x}_d|\boldsymbol{\mu}_j)}$.

   In M-step, set the derivative with respect to π_n to 0, we obtain $\pi_n = \frac{\sum_{d=1}^{D} \gamma(z_{dn})}{D}$. Set the derivative with respect to $\boldsymbol{\mu_n}$ to 0, we obtain $\boldsymbol{\mu_n} = \frac{\sum_{d=1}^{D} \gamma(z_{dn})\mathbf{x_d}}{\sum_{d=1}^{D} \gamma(z_{dn})}$.

7. Hoeffding's inequality is a powerful technique—perhaps the most important inequality in learning theory for bounding the probability that sums of bounded random variables are too large or too small. Below are some related inequalities you are required to provide proof:

   (a) **(Markov's inequality).** Let $Z \geq 0$ be a non-negative random variable. Then for all $t \geq 0$, show that

   $$\mathbb{P}(Z \geq t) \leq \frac{\mathbb{E}(Z)}{t}, \tag{3}$$

   where $\mathbb{E}$ denotes the expectation operator. (6 points)

   (b) **(Chebyshev's inequality).** Let $Z \geq 0$ be a random variable with $\mathrm{Var}(Z) < \infty$. Show that

   $$\mathbb{P}(Z \geq \mathbb{E}[Z] + t \text{ or } Z \leq \mathbb{E}[Z] - t) \leq \frac{\mathrm{Var}(Z)}{t^2}, \qquad \text{for } t \geq 0, \tag{4}$$

   where $\mathrm{Var}(Z)$ denotes the variance of Z. (6 points)

   Solution:

   (a) *Proof.* We note that $\mathbb{P}(Z \geq t) = \mathbb{E}[\mathbf{1}\{Z \geq t\}]$, and that if $Z \geq t$, then it must be the case that $Z/t \geq 1 \geq \mathbf{1}\{Z \geq t\}$, while of $Z < t$, then we still have $Z/t \geq 0 = \mathbf{1}\{Z \geq t\}$. Thus

   $$\mathbb{P}(Z \geq t) = \mathbb{E}[\mathbf{1}\{Z \geq t\}] \leq \mathbb{E}\left[\frac{Z}{t} = \frac{\mathbb{E}[Z]}{t}\right],$$

   as desired. □

   (b) *Proof.* The result is an immediate consequence of Markov's inequality. We note that either $Z \geq \mathbb{E}(Z)+t$ or $Z \leq \mathbb{E}[Z] - t$, we have $(Z - \mathbb{E}(Z))^2 \geq t^2$. Thus,

   $$\begin{aligned}\mathbb{P}(Z \geq \mathbb{E}[Z] + t \text{ or } Z \leq \mathbb{E}[Z] - t) &= \mathbb{P}((Z - \mathbb{E}(Z))^2 \geq t^2) \\ &\leq \frac{\mathbb{E}[(Z - \mathbb{E}(Z))^2]}{t^2} = \frac{\mathrm{Var}(Z)}{t^2},\end{aligned}$$

   where the inequality holds due to the Markov's inequality. □

8. Recall that to show VC dimension is d for hypotheses $\mathcal{H}$ can be done via showing that VC dim$(\mathcal{H}) \leq d$ and VC dim$(\mathcal{H}) \geq d$. More specifically, to prove that VC dim$(\mathcal{H}) \geq d$ it suffices to give d examples that can be shattered; to prove VC dim$(\mathcal{H}) \leq d$ one must show that no set $d+1$ examples can be shattered.

   For each one of the following function classes, find the VC dimension. State your reasoning based on the presented hint above. (Note that: solutions with the correct answer but without adequate explanation will not earn marks.)

   (a) **Halfspaces in** $\mathbb{R}^2$. Examples lying in or on the halfspace are labeled $+1$, and the remaining examples are labeled -1. (3 points)

   (b) **Axis-parallel rectangles in** $\mathbb{R}^2$. Points lying on or inside the target rectangle are labeled $+1$, and points lying outside the target rectangle are labeled -1. (3 points)

   (c) **Closed sets in** $\mathbb{R}^2$. All points lying in the set or on the boundary of the set are labeled $+1$, and all points lying outside the set are labeled -1. (3 points)

   (d) How many training examples suffice to assure with probability 0.9 that a consistent learner using the function classes presented in (b) will learn the target function with accuracy of at least 0.95? (4 points) (Hint: we use the following bounds on sample complexity: $m \geq \frac{1}{\epsilon}(4\log_2(2/\delta)+8\text{VC dim}(\mathcal{H})\log_2(13/\epsilon))$).

   Solution:

   (a) It is easily shown that any three non-collinear points (e.g., $(0,1)$, $(0,0)$, $(1,0)$) are shattered by $\mathcal{H}$. Thus, VC dim$(\mathcal{H}) \geq 3$. We now show that no set of size four can be shattered by $\mathcal{H}$. If at least three of the points are collinear then there is no halfspace that contains the two extreme points but does not contain the middle points. Thus the four points cannot be shattered if any three are collinear. Next, suppose that the points form a quadrilateral. There is no halfspace which labels one pair of diagonally opposite points positive and the other pair of diagonally opposite points negative. The final case is that one point p is in the triangle defined by the other three. In this case there is no halfspace which labels p differently from the other three. Thus clearly the four points cannot be shattered. Therefore we have demonstrated that VC dim$(\mathcal{H}) = 3$.

(b) First, it is easily seen that there is a set of four points (e.g., $(0,1)$, $(0,-1)$, $(1,0)$, $(-1,0)$) that can be shattered. Thus VC dim$(\mathcal{H}) \geq 4$. We now argue that no set of five points can be shattered. The smallest bounding axis-parallel rectangles defined by the five points is in fact defined by at most four of the points. For p a non-defining point in the set, we see that the set cannot be shattered since it is not possible for p to be classified as negative while also classifying the others as positive. Thus VC dim$(\mathcal{H}) = 4$.

(c) Any set can be shattered by $\mathcal{H}$, since a closed set can assume any shape in $\mathbb{R}^n$. Thus, the largest set that can be shattered by $\mathcal{H}$ is infinite, and hence VC dim$(\mathcal{H}) = \infty$.

(d) The bound is $m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8\text{VC dim}(\mathcal{H})\log_2(13/\epsilon))$. Then just by plugging in the numbers (VC dim$(\mathcal{H}) = 4$, $\delta = 0.1$ and $\epsilon = 0.05$), we have $m \geq 5480$.

Optimization and Machine Learning, Spring 2020
Homework 4
(Due Tuesday, May 12 at 11:59pm (CST))

1. Given a training dataset $S = \{(x_i, y_i)\}_{i=1}^n$, in which $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ denote the i-th sample and the i-th label, respectively. Suppose that we use S to train a machine learning model based on Adaboost. At the end of the t-th iteration ($t = 1, 2, ..., T$), the importance of the i-th ($i = 1, 2, ..., n$) sample x_i is reweighted as

$$D_i^{(t+1)} = D_i^{(t)} \exp\left(-\alpha_t y_i h_t(x_i)\right),$$

where α_t is the weight of the t-th weakly binary classifier h_t, i.e.,

$$\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t}), \text{ with } \epsilon_t = \sum_{i=1}^{n} D_i^{(t)} \mathbb{1}(y_i \neq h_t(x_i)).$$

To classify an arbitrary test sample x, we calculate $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ and then return its sign. Now let's show that if every learner h_t ($\forall t$) achieves 51% classification accuracy (that is, only slightly better than random guessing), AdaBoost will converge to zero training error.

   (a) Let's change the update rule so that the weights of each iteration are normalized, that is, $\sum_{i=1}^{n} D_i^{(t)} = 1$ ($\forall t$). In this sense, we can treat the weights as a discrete probability distribution over the sample points. Hence we rewrite the update rule by

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$

   where Z_t is the normalization factor, $\forall t$. Please show that the following formula is satisfied,

$$Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}.$$

   (5 points)
   Solution:

$$\begin{aligned} Z_t &= \sum_{i=1}^{n} D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i)) \\ &= \sum_{y_i \neq h_t(x_i)} D_i^{(t)} \exp(\alpha) + \sum_{y_i = h_t(x_i)} D_i^{(t)} \exp(-\alpha) \\ &= \epsilon_t (\frac{1-\epsilon_t}{\epsilon_t})^{1/2} + (1-\epsilon_t)(\frac{1-\epsilon_t}{\epsilon_t})^{-1/2} \\ &= 2\sqrt{\epsilon_t(1-\epsilon_t)} \end{aligned}$$

   (b) Assume that the initial weights follow uniform distribution, i.e.,

$$D_1^{(1)} = D_2^{(1)} = \cdots = D_n^{(1)} = \frac{1}{n}.$$

   Please show that

$$D_i^{(t)} = \frac{1}{n \prod_{t=1}^{T} Z_t} e^{-y_i f(x_i)},$$

   where $i = 1, 2, ..., n$ and $t = 2, 3, ..., T$. (5 points)
   Solution:

$$
\begin{aligned}
D_i^{(T)} &= \frac{D_i^{(T-1)} \exp\left(-\alpha_{T-1} y_i h_{T-1}(x_i)\right)}{Z_{T-1}} \\
&= \frac{D_i^{(T-2)} \exp\left(-\alpha_{T-1} y_i h_{T-1}(x_i)\right) \exp\left(-\alpha_{T-2} y_i h_{T-2}(x_i)\right)}{Z_{T-1} Z_{T-2}} \\
&\cdots \\
&= \frac{D_i^{(1)} e^{-y_i \sum_{t=1}^{T-1} \alpha_t h_t(x_i)}}{\prod_{t=1}^{T-1} Z_t} \\
&= \frac{e^{-y_i f(x_i)}}{n \prod_{t=1}^{T-1} Z_t}
\end{aligned}
$$

(c) Let m be the number of sample points that Adaboost classifies incorrectly. Please show that

$$
\sum_{i=1}^{n} e^{-y_i f(x_i)} \geq m.
$$

(5 points)
Solution:

$$
\begin{aligned}
\sum_{i=1}^{n} e^{-y_i f(x_i)} &= \sum_{y=f(x_i)} e^{-1} + \sum_{y \neq f(x_i)} e^{1} \\
&= me + (n-m)e^{-1} \\
&= ne^{-1} + (e - e^{-1})m \\
&\geq m
\end{aligned}
$$

(d) Based on the results in (a), (b), and (c), please show that once $\epsilon_t \leq 0.49$ is satisfied for every learner h_t ($\forall t$), then we have $m \to 0$ as $T \to \infty$. (5 points)
Solution:

$$
\begin{aligned}
&Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)} < 1 \\
&T \to \infty, \sum_{i=1}^{n} e^{-y_i f(x_i)} = n \prod_{t=1}^{T-1} Z_t \to 0 \\
&m \leq \sum_{i=1}^{n} e^{-y_i f(x_i)} \to 0
\end{aligned}
$$

2. Suppose that we are interested in learning a classifier, such that at any turn of a game we can pose a question, like "should I attack this ant hill now?", and get an answer. That is, we want to build a classifier which we can feed some features on the current game state, and get the output "attack" or "don't attack". There are many possible ways to define what the action "attack" means, but for now let's define it as sending all friendly ants that can see the ant hill under consideration towards it.

   Let's recall the AdaBoost algorithm described in class. Its input is a dataset $\{(x_i, y_i)\}_{i=1}^{n}$, with x_i being the i-th sample, and $y_i \in \{-1, 1\}$ denoting the i-th label, $i = 1, 2, ..., n$. The features might be composed of a count of the number of friendly ants that can see the ant hill under consideration, and a count of the number of enemy ants these friendly ants can see. For example, if there were 10 friendly ants that could see a particular ant hill, and 5 enemy ants that the friendly ants could see, we would have:

$$
x_1 = \begin{bmatrix} 10 \\ 5 \end{bmatrix}.
$$

   The label of the example x_1 is $y_1 = 1$, once the friendly ants were successful in razing the enemy ant hill, and $y_1 = 0$ otherwise. We could generate such examples by running a greedy bot (or any other opponent bot) against a bot that we make periodically try to attack an enemy ant hill. Each time this bot tries the attack, we record (say, after 20 turns or some other significant amount of time) whether the attack was successful or not.

(a) Let ϵ_t denote the error of a weak classifier h_t:

$$\epsilon_t = \sum_{i=1}^{n} D_i^{(t)} \mathbb{1}(y_i \neq h_t(x_i)).$$

In the simple "attack" / "don't attack" scenario, suppose that we have implemented the following six weak classifiers:

$$h^{(1)}(x_i) = 2 * \mathbb{1}(x_{i1} \geq 2) - 1, \qquad h^{(4)}(x_i) = 2 * \mathbb{1}(x_{i2} \leq 2) - 1,$$
$$h^{(2)}(x_i) = 2 * \mathbb{1}(x_{i1} \geq 5) - 1, \qquad h^{(5)}(x_i) = 2 * \mathbb{1}(x_{i2} \leq 5) - 1,$$
$$h^{(3)}(x_i) = 2 * \mathbb{1}(x_{i1} \geq 10) - 1, \qquad h^{(6)}(x_i) = 2 * \mathbb{1}(x_{i2} \leq 10) - 1.$$

Given ten training data points ($n = 10$) as shown in Fig. 1, please show that what is the minimum value

i	x_{i1}	x_{i2}	y_i
1	3.5	0.5	1
2	2.5	1.5	1
3	5.5	5.5	1
4	7.5	7.5	1
5	6.5	9.5	1
6	1.5	2.5	-1
7	4.5	4.5	-1
8	3.5	7.5	-1
9	1.5	11.5	-1
10	6.5	11.5	-1

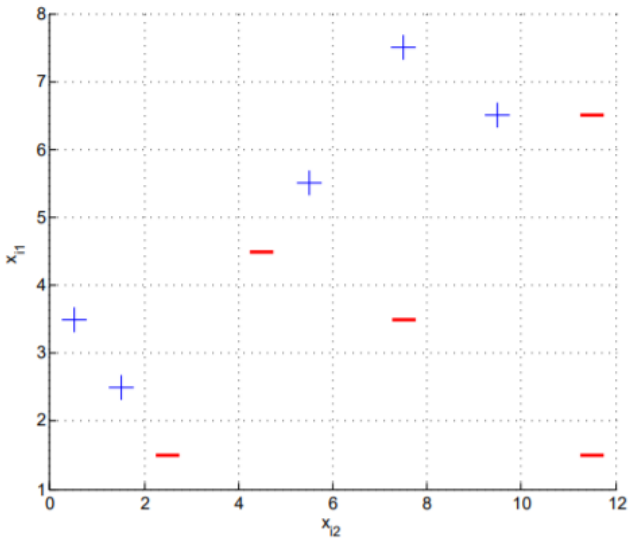


Figure 1: The training data in (a).

of ϵ_1 and which of $h^{(1)}, ..., h^{(6)}$ achieve this value? Note that there may be multiple classifiers that all have the same ϵ_1. You should list all classifiers that achieve the minimum ϵ_1 value. (5 points)
Solution:
The value of ϵ_1 for each of the classifiers is: $3/10, 3/10, 5/10, 3/10, 5/10$, and $3/10$. So, the minimum value is $3/10$ and classifiers 1, 2, 4, and 6 achieve this value.

(b) For all the questions in the remainder of this section, let h_1 denote $h^{(1)}$ chosen in the first round of boosting. (That is, $h^{(1)}$ was the classifier that achieved the minimum ϵ_1.)

(1) What is the value of α_1 (the weight of this first classifier h_1)? Keep in mind that the log in the formula for α_t is a natural log (base e). (5 points)
Solution:
Plugging into the formula for α we get: $\alpha_1 = \frac{1}{2} \log \frac{1 - \epsilon_1}{\epsilon_1} = \frac{1}{2} \log \frac{7}{3} = 0.4236$

(2) What should Z_t be in order to make sure the distribution $D^{(t+1)}$ is normalized correctly? That is, derive the formula of Z_t in terms of $D^{(t)}$, α_t, h_t, and $\{(x_i, y_i)\}_{i=1}^n$, that will ensure $\sum_{i=1}^n D_i^{(t+1)} = 1$. (5 points)
Solution:

$$Z_t = \sum_{k=1}^{n} D_T(k) \exp(-\alpha_t y_k h_t(x_k))$$

(3) Which points will increase in significance in the second round of boosting? That is, for which points will we have $D_i^{(1)} < D_i^{(2)}$? What are the values of $D^{(2)}$ for these points? (5 points)

Solution:
The points that $h(1)$ misclassifies will increase in weight. These are the points $i = 7, 8, 10$ from the data table. Their new weight under D_2 will be:

$$\begin{aligned} D_2(i) &= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i)}{Z_t} \\ &= \frac{\exp\{0.4236\}}{3 * \exp\{0.4236\} + 7 * \exp\{-0.4236\}} \\ &= \frac{1}{6} \end{aligned}$$

(4) In the second round of boosting, the weights on the points will be different, and thus the error ϵ_2 will also be different. Which of $h^{(1)}, ..., h^{(6)}$ will minimize ϵ_2? (Which classifier will be selected as the second weak classifier h_2?) What is its value of ϵ_2? (5 points)
Solution:
$h^{(4)}$ will be chosen.

Classifier	ϵ_2
$h^{(1)}$	$1/2$
$h^{(2)}$	$1/6 + 2/14 = 13/42$
$h^{(3)}$	$5/14 = 0.3571$
$h^{(4)}$	$3/14$
$h^{(5)}$	$1/6 + 4/14 = 19/42$
$h^{(6)}$	$2/6 + 1/14 = 17/42$

(5) What will the average error of the final classifier H be, if we stop after these two rounds of boosting? That is, if $H(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x))$, what will the training error $\epsilon = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(y_i \neq h(x_i))$ be? Is this more, less, or the same as the error we would get, if we just used one of the weak classifiers instead of this final classifier H? (5 points)
Solution:
The classifier after two rounds is:

$$h(x) = sign(0.5\log(7/3)h^{(1)}(x) + 0.5\log(11/3)h^{(4)}(x))$$

Since $log(11/3) > log(7/3)$ the classifier h will always go with the guess made by $h(4)$. So, it will not do any better than the error we could get using a single weak classifier, $\epsilon = 3/10$. More rounds of boosting are necessary before the interplay of specific settings of the α becomes relevant and allows us to do better than a single weak classifier.

3. Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, please verify the following new kernels will also be valid:

(a) $k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})k_1(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')$, where $f(\cdot)$ is any function. (2 points)
(b) $k(\mathbf{x}, \mathbf{x}') = q(k_1(\mathbf{x}, \mathbf{x}'))$, where $q(\cdot)$ is a polynomial with nonnegative coefficients. (3 points)
(c) $k(\mathbf{x}, \mathbf{x}') = \exp(k_1(\mathbf{x}, \mathbf{x}'))$. (5 points)
(d) $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{A}\mathbf{x}'$, where $\mathbf{A}$ is a symmetric positive semi-definite matrix. (5 points)

Solution:

(a) Since $k_1(\mathbf{x}, \mathbf{x}')$ is a valid kernel, there must exist a feature vector $\phi(\mathbf{x})$ such that

$$k_1(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}').$$

Then we can rewrite the given kernel as

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x})\phi(\mathbf{x})^\top \phi(\mathbf{x}')f(\mathbf{x}') \\ &= \mathbf{v}(\mathbf{x})^\top \mathbf{v}(\mathbf{x}'), \end{aligned}$$

where $\mathbf{v}(\mathbf{x}) \triangleq f(\mathbf{x})\phi(\mathbf{x})$. We can see that the kernel can be rewritten as the scalar product of feature vectors, and hence is a valid kernel.

(b) Suppose $q(x) = \sum_{i=1}^{n} a_n x^n, \forall a_n \geq 0$, then the kernel can be expressed as

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{n} a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n .$$

We focus on the i-th term of the kernel, which is $a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n$. Since $k_1(\mathbf{x}, \mathbf{x}')$ is a valid kernel, the product of kernels is also a valid kernel. Hence, $a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n$ is a valid kernel. With the fact that the sum of kernels is a valid kernel, the original kernel is valid.

(c) Let $\mathbf{K}$ be the Gram matrix. The (i, j)-th entry of $\mathbf{K}$ is defined by $\mathbf{K}_{i,j} \triangleq k(\mathbf{x}_i, \mathbf{x}_j)$. Since $k_1(\mathbf{x}, \mathbf{x}')$ is a valid kernel, we have $k_1(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}', \mathbf{x})$. Hence, the Gram matrix $\mathbf{K}$ is symmetric. In addition, $k(\mathbf{x}, \mathbf{x}')$ is an exponential function, which leads to $k(\mathbf{x}, \mathbf{x}')$ is always greater than zero. Therefore, the Gram matrix $\mathbf{K}$ is positive definite. Applying, Mercer's condition, $k(\mathbf{x}, \mathbf{x}')$ is a valid kernel.

(d) Since $\mathbf{A}$ is a symmetric positive semi-definite matrix, we can decompose $\mathbf{A}$ as $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix. When $\mathbf{A}$ is positive semi-definite, the entries of $\mathbf{\Lambda}$ are nonnegative. Hence, we can rewrite the kernel as

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^\top \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top \mathbf{x}' \\ &= (\mathbf{\Lambda}^{1/2}\mathbf{Q}^\top \mathbf{x})^\top (\mathbf{\Lambda}^{1/2}\mathbf{Q}^\top \mathbf{x}') \\ &= \mathbf{\Phi}(\mathbf{x})^\top \mathbf{\Phi}(\mathbf{x}'), \end{aligned}$$

where $\mathbf{\Phi}(\mathbf{x}) \triangleq \mathbf{\Lambda}^{1/2}\mathbf{Q}^\top \mathbf{x}$. We can see that the kernel can be rewritten as the scalar product of feature vectors, and hence is a valid kernel.

4. Consider the space of all possible subsets A of a given fixed set D. Show that the kernel function $k(A_1, A_2) = 2^{|A_1 \cap A_2|}$ corresponds to an inner product in a feature space of dimensionality $2^{|D|}$ defined by the mapping $\phi(A)$ where A is a subset of D and the element $\phi_U(A)$, indexed by the subset U, is given by

$$\phi_U(A) = \begin{cases} 1, & \text{if } U \subseteq A; \\ 0, & \text{otherwise.} \end{cases}$$

Here $U \subseteq A$ denotes that U is either a subset of A or is equal to A. (10 points)

Solution:
First of all, we consider the case of $A = D$. In that case, $\phi(D)$ must be defined. This will map to a vector $2^{|D|}$ 1s, one for each possible subset of D, including D itself as well as the empty set. Now we consider another case of $A \subset D$, $\phi(A)$ will have 1s in all positions that correspond to subsets of A and 0s in all other positions. Therefore, $\phi(A_1)^\top \phi(A_2)$ will count the number of subsets shared by A_1 and A_2. However, this can just as well be obtained by counting the number of elements in the intersection of A_1 and A_2, and then raising 2 to this number, which is exactly $2^{|A_1 \cap A_2|}$ does.

5. Suppose we have a data set of input vectors $\{\mathbf{x}_n\}$ with corresponding target values $t_n \in \{-1, 1\}$, and suppose that we model the density of input vectors within each class separately using a Parzen kernel density estimator which is defined as follows

$$p(\mathbf{x}|t) = \frac{1}{N_t} \sum_{n=1}^{N} \frac{1}{Z_k} k\left(\mathbf{x}, \mathbf{x}_n\right) \delta(t, t_n)$$

where $k(\mathbf{x}, \mathbf{x}')$ is a valid kernel, Z_k is the normalization constant for the kernel and $\delta(t, t_n)$ equals 1 if $t = t_n$ and 0 otherwise.

(a) Write down the minimum misclassification-rate decision rule assuming the two classes have equal prior probability. (3 points)

(b) Show that, if the kernel is chosen to be $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$, then the classification rule reduces to simply assigning a new input vector to the class having the closest mean. (4 points)

(c) Show that, if the kernel takes the form $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$, that the classification is based on the closest mean in the feature space $\phi(\mathbf{x})$. (4 points)

Solution:

(a) From Bayes' theorem we have

$$p(t|\mathbf{x}) \propto p(\mathbf{x}|t)p(t)$$

where, from the stem,

$$p(\mathbf{x}|t) = \frac{1}{N_t}\sum_{n=1}^{N}\frac{1}{Z_k}k\left(\mathbf{x},\mathbf{x}_n\right)\delta\left(t,t_n\right).$$

The minimum misclassification-rate is achieved if, for each new input vector, $\tilde{\mathbf{x}}$, we chose $\tilde{t}$ to maximize $p(\tilde{t}|\tilde{\mathbf{x}})$. With equal class priors, this is equivalent to maximizing $p(\tilde{\mathbf{x}}|\tilde{t})$ and thus

$$\tilde{t} = \begin{cases} +1 & \text{iff } \frac{1}{N_{+1}}\sum_{i:t_i=+1}k\left(\tilde{\mathbf{x}},\mathbf{x}_i\right) \geqslant \frac{1}{N_{-1}}\sum_{j:t_j=-1}k\left(\tilde{\mathbf{x}},\mathbf{x}_j\right) \\ -1 & \text{otherwise} \end{cases}$$

Here we have dropped the factor $1/Z_k$ since it only acts as a common scaling factor. Using the encoding scheme for the label, this classification rule can be written in the more compact form

$$\tilde{t} = \text{sign}\left(\sum_{n=1}^{N}\frac{t_n}{N_{t_n}}k\left(\tilde{\mathbf{x}},\mathbf{x}_n\right)\right).$$

(b) Now we take $k(\mathbf{x},\mathbf{x}') = \mathbf{x}^\top\mathbf{x}_n$, which results in the kernel density

$$p(\mathbf{x}|t=+1) = \frac{1}{N_{+1}}\sum_{n:t_n=+1}\mathbf{x}^{\mathrm{T}}\mathbf{x}_n = \mathbf{x}^{\mathrm{T}}\bar{\mathbf{x}}^{+}.$$

Here, the sum in the middle expression runs over all vectors $\mathbf{x}_n$ for which $t_n = +1$ and $\bar{x}^+$ denotes the mean of these vectors, with the corresponding definition for the negative class. Note that this density is improper, since it cannot be normalized. However, we can still compare likelihoods under this density, resulting in the classification rule

$$\tilde{t} = \begin{cases} +1 & \text{if } \tilde{\mathbf{x}}^{\mathrm{T}}\bar{\mathbf{x}}^{+} \geqslant \tilde{\mathbf{x}}^{\mathrm{T}}\bar{\mathbf{x}}^{-} \\ -1 & \text{otherwise} \end{cases}$$

(c) The same argument in (a) could also apply that the feature space is $\phi(\mathbf{x})$.

6. The problem of maximizing margin can be converted into an following equivalent problem

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{argmin}} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to } t_n(\mathbf{w}^\top\phi(\mathbf{x}_n)+b) \geq 1, \quad n = 1,\cdots,N. \end{aligned}$$

where $\phi(\mathbf{x})$ is a fixed feature-space transformation.

(a) By introducing Lagrange multipliers $\{a_n\}$, please give the Lagrangian function and the dual representation of the maximum margin problem. (8 points)

(b) Please show that the value ρ of the margin for the maximum-margin hyperplane is given by

$$\frac{1}{\rho^2} = \sum_{n=1}^{N} a_n.$$

(Hint: $\{a_n\}$ can be obtained by solving the dual representation of the maximum margin problem.) (6 points)

Solution:

(a) The Lagrangian function is given by

$$L(\mathbf{w},b,\mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N}a_n\left\{t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_n\right)+b\right)-1\right\}. \tag{1}$$

The dual representation of the maximum margin problem is given by

$$\max_{\mathbf{a}} \widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{2}$$

$$\text{subject to } a_n \geq 0, \quad n = 1, \cdots, N, \tag{3}$$

$$\sum_{n=1}^{N} a_n t_n = 0, \tag{4}$$

where $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$.

(b) Let the value of the margin ρ be $1/\|\mathbf{w}\|$ and so $1/\rho^2 = \|\mathbf{w}\|^2$. From the KKT conditions of the dual problem which is

$$\begin{aligned} a_n &\geqslant 0 \\ t_n y(\mathbf{x}_n) - 1 &\geqslant 0 \\ a_n \{t_n y(\mathbf{x}_n) - 1\} &= 0, \end{aligned}$$

we see that, for the maximum margin solution, the second term of (1) vanishes and so we have

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2. \tag{5}$$

By setting the derivatives of (1) with respect to $\mathbf{w}$ and b equal to zero, we obtain the following two conditions

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \tag{6}$$

$$0 = \sum_{n=1}^{N} a_n t_n. \tag{7}$$

Using (5) together with (6), the dual (2) can be written as

$$\frac{1}{2}\|\mathbf{w}\|^2 = \sum_{n}^{N} a_n - \frac{1}{2}\|\mathbf{w}\|^2,$$

from which the desired result follows.

Optimization and Machine Learning, Spring 2020
Homework 5
(Due Tuesday, June 2 at 11:59pm (CST))

1. Show that weighted Euclidean distance in $\mathbb{R}^p$,

$$d_e^{(w)}(x_i, x_{i'}) = \frac{\sum_{l=1}^p w_l (x_{il} - x_{i'l})^2}{\sum_{l=1}^p w_l},$$

satisfies

$$d_e^{(w)}(x_i, x_{i'}) = d_e(z_i, z_{i'}) = \sum_{l=1}^p (z_{il} - z_{i'l})^2,$$

where

$$z_{il} = x_{il} \left(\frac{w_l}{\sum_{l=1}^p w_l} \right)^{1/2}.$$

Thus weighted Euclidean distance based on x is equivalent to unweighted Euclidean distance based on z. (15 points)

Solution:

$$\begin{aligned}
d_e^{(w)}(x_i, x_i') &= \frac{\sum_{l=1}^p w_l (x_{il} - x_{i'l})^2}{\sum_{l=1}^p w_i} \\
&= \sum_{l=1}^p \frac{w_l}{\sum_{l=1}^p w_l} (x_{il} - x_{i'l})^2 \\
&= \sum_{l=1}^p \left(\frac{w_l}{\sum_{l=1}^p w_l} (x_{il} - x_{i'l}) \right)^2 \\
&= \sum_{l=1}^p \left(\frac{w_l}{\sum_{l=1}^p w_l}^{1/2} x_{il} - \frac{w_l}{\sum_{l=1}^p w_l}^{1/2} x_{i'l} \right)^2 \\
&= \sum_{l=1}^p (z_{il} - z_{i'l})^2 \\
&= d_e(z_i, z_{i'})
\end{aligned}$$

2. Consider a dataset of n observations $\mathbf{X} \in \mathbb{R}^{n \times d}$, and our goal is to project the data onto a subspace having dimensionality p, $p < d$. Prove that PCA based on projected variance maximization is equivalent to PCA based on projected error (Euclidean error) minimization. (20 points)

Solution: Suppose $\mathbf{X}$ has been centralized. Let $\mathbf{V} \in \mathbb{R}^{d \times p}$ reprensent the projected matrix and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Then we have two different optimization goals. The one based on projected variance maximization is

$$\text{maximize} \quad \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = Tr(\mathbf{V}\mathbf{V}^T\mathbf{X}^T\mathbf{X}\mathbf{V}\mathbf{V}^T) = Tr(\mathbf{X}^T\mathbf{X}\mathbf{V}\mathbf{V}^T)$$

, the other one based on projected error minimization is

$$\text{minimize} \quad \|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = Tr((\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T))$$

And we have

$$\begin{aligned}
Tr((\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T)) &= Tr((\mathbf{I} - \mathbf{V}\mathbf{V}^T)^T\mathbf{X}^T\mathbf{X}(\mathbf{I} - \mathbf{V}\mathbf{V}^T)) \\
&= Tr(\mathbf{X}^T\mathbf{X}(\mathbf{I} - \mathbf{V}\mathbf{V}^T)) \\
&= Tr(\mathbf{X}^T\mathbf{X}) - Tr(\mathbf{X}^T\mathbf{X}\mathbf{V}\mathbf{V}^T)
\end{aligned}$$

Since $Tr(\mathbf{X}^T\mathbf{X})$ is a constant value, so projected variance maximization is equivalent to projected error minimization.

3. Show that the conventional linear PCA algorithm is recovered as a special case of kernel PCA if we choose the linear kernel function given by $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$. (15 points)

   Solution: Suppose $\mathbf{X}$ has been centralized. The kernel function is given by $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$, so we have the matrix form $\mathbf{K} = \mathbf{X}^T\mathbf{X}$. We can represent principal components $\mathbf{v} = \sum_{i=1}^{n} a_i\mathbf{x}_i = \mathbf{X}\alpha$. And to solve the conventional linear PCA, we have

$$\begin{aligned} & \frac{1}{n}\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda\mathbf{v} \\ \Rightarrow \quad & \mathbf{X}^T\frac{1}{n}\mathbf{X}\mathbf{X}^T\mathbf{v} = \mathbf{X}^T\lambda\mathbf{v} \\ \Rightarrow \quad & \frac{1}{n}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{X}\alpha = \lambda\mathbf{X}^T\mathbf{X}\alpha \\ \Rightarrow \quad & \frac{1}{n}\mathbf{K}\mathbf{K}\alpha = \lambda\mathbf{K}\alpha \\ \Rightarrow \quad & \frac{1}{n}\mathbf{K}\alpha = \lambda\alpha \end{aligned}$$

   Then we show the conventional linear PCA is a special case of kernel PCA with the linear kernel function given by $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$.

4. Let $S = \{x^{(1)}, \cdots, x^{(n)}\}$ be a dataset of n samples with 2 features, i.e $x^{(i)} \in \mathbb{R}^2$. The samples are classified into 2 categories with labels $y^{(i)} \in \{0, 1\}$. A scatter plot of the dataset is shown in Figure 1.

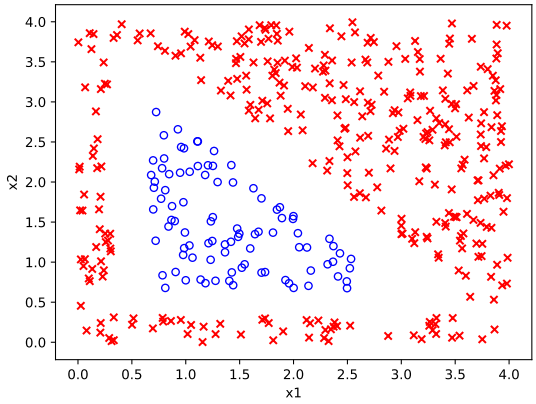


Figure 1: Plot of dataset S.

   The examples in class 1 are marked as as "×" and examples in class 0 are marked as "○". We want to perform binary classification using a simple neural network with the architecture shown in Figure 2.

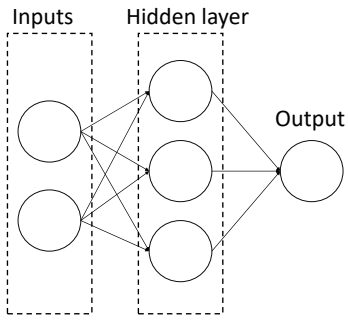


Figure 2: Architecture for our simple neural network.

   Denote the two features x_1 and x_2, the three neurons in the hidden layer h_1, h_2, and h_3, and the output neuron as o. Let the weight from x_i to h_j be $w_{i,j}^{[1]}$ for $i \in \{1, 2\}, j \in \{1, 2, 3\}$, and the weight from h_j to o be

$w_j^{[2]}$. Finally, denote the intercept weight for h_j as $w_{0,j}^{[1]}$, and the intercept weight for o as $w_0^{[2]}$. For the loss function, we'll use average squared loss instead of the usual negative log-likelihood:

$$l = \frac{1}{n}\sum_{i=1}^{n}(o^{(i)} - y^{(i)})^2,$$

where $o^{(i)}$ is the result of the output neuron for example i.

(a) Suppose we use the sigmoid function as the activation function for h_1, h_2, h_3 and o. What is the gradient descent update to $w_{2,1}^{[1]}$, assuming we use a learning rate of α? Your answer should be written in terms of $x^{(i)}$, $o^{(i)}$, $y^{(i)}$, and the weights. (10 points)

Solution: Let z^{h_i} for $i \in \{1, 2\}$ and z^o be the input to the sigmoid function at the hidden and output layers.

$$\begin{aligned}\frac{\partial l}{\partial w_{2,1}^{[1]}} &= \frac{\partial l}{\partial o}\frac{\partial o}{\partial z^o}\frac{\partial z^o}{\partial h_1}\frac{\partial h_1}{\partial z^{h_1}}\frac{\partial z^{h_1}}{\partial w_{2,1}^{[1]}} \\ &= \frac{1}{n}\sum_{i=1}^{n} 2o^i(o^i - y^i)(1 - o^i)w_1^{[2]}h_1(1 - h_1)x_2^{(i)},\end{aligned}$$

where we exploit the fact that the derivative (w.s.t. z) of the sigmoid function is $\frac{\partial \sigma}{\partial z} = \sigma(z)(1 - \sigma(z))$.

(b) Now, suppose instead of using the sigmoid function for the activation function for h_1, h_2, h_3 and o, we instead used the step function $f(x)$, defined as

$$f(x) = \begin{cases} 1, x \geq 0, \\ 0, x < 0. \end{cases}$$

Is it possible to have a set of weights that allow the neural network to classify this dataset with 100% accuracy? If it is possible, please provide a set of weights that enable 100% accuracy and explain your reasoning for those weights in your PDF. If it is not possible, please explain your reasoning in your PDF. (10 points) (Hint: There are three sides to a triangle, and there are three neurons in the hidden layer.)

Solution: Based on the hinted triangle, we construct the following matrix multiplication as

$$\begin{bmatrix} -1 & 5 & 0 \\ -1 & 0 & 5 \\ 4.2 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}. \tag{1}$$

When the data point is within the triangle, $f(z) = [1, 1, 1]^T$. Otherwise, it's one of the seven binary vectors (e.g., $[0, 1, 0]^T$).

Then, we construct the second matrix multiplication as

$$\begin{bmatrix} -1 & -1 & -1 & -2.5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{2}$$

Note that, the 1 in the first row of the all-one vector is the bias. Therefore, only when a data point is inside the triangle, it produces the result less than 0, and hence categorized as class 0. Otherwise, it will be classified as class 1. Hence, the accuracy is 100%.

(c) Let the activation functions for h_1, h_2, h_3 be the linear function $f(x) = x$ and the activation function for o be the same step function as before. Is it possible to have a set of weights that allow the neural network to classify this dataset with 100% accuracy? If it is possible, please provide a set of weights that enable 100% accuracy and explain your reasoning for those weights in your PDF. If it is not possible, please explain your reasoning in your PDF. (10 points)

Solution: No, it is impossible. Because the current classes cannot be linearly separated. If the activation is linear, then it's an affine transformation from 2D to 3D spaces, and the classification problem will still be non-linearly in the 3D space, which cannot be solved by a step function.

5. Convolutional neural networks targets the processing of 2-D features instead of the 1-D ones in multi-layer perceptron (MLP), the structure of which is depicted in Fig. 3.

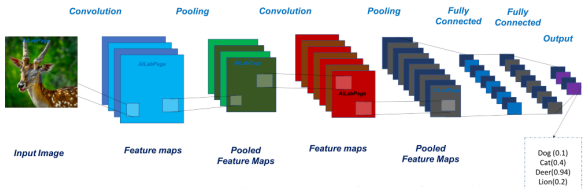


Figure 3: https://mc.ai/how-does-convolutional-neural-network-work/

(a) Kernel convolution is process where we take a kernel (or filter), we pass it over our image and transform it based on the values from filter. Now, you are given the following formula

$$G(m,n) = (f * h)(m,n) = \sum_j \sum_k h(j,k) f(m-j, n-k),$$

where the input image is denoted by f and the kernel by h. The indexes of rows and columns of the result matrix are marked with m and n respectively. Please calculate the feature maps, if you are given the following 3×3 image matrix and 2×2 kernel matrix. (5 points)

1	2	3
4	5	6
7	8	9

Table 1: 3×3-Image Matrix.

1	0
0	1

Table 2: 2×2-Kernel Matrix.

Solution: By the formula, we have the feature maps, as shown in Tab. 3.

6	8
12	14

Table 3: 2×2-feature maps.

(b) Assume the input with the size of (width = 28, height = 28) and a filter with the size of (width =5, height = 5) and the convolutional layer parameters are $S = 1$ (the stride), $P = 0$ (the amount of zero padding). What is the exact size of the convolution output? (5 points)

Solution: The Conv layer:

- Accepts the input with the size 28×28
- Requires the hyperparameters:
  - A filter with the size 5×5,
  - the stride $S = 1$,
  - the amount of zero padding $P = 0$.
- Produces a output of size $W_{\text{out}} \times H_{\text{out}}$ where
  - $W_{\text{out}} = (28 - 5 + 0)/1 + 1 = 24$
  - $H_{\text{out}} = (28 - 5 + 0)/1 + 1 = 24$.

6. Consider the following grid environment. Starting from any unshaded square, you can move up, down, left, or right. Actions are deterministic and always succeed (e.g. going left from state 1 goes to state 0) unless they will cause the agent to run into a wall. The thicker edges indicate walls, and attempting to move in the direction of a wall results in staying in the same square. Taking any action from the green target square (no. 5) earns a reward of +5 and ends the episode. Taking any action from the red square of death (no. 11) earns a reward of -5 and ends the episode. Otherwise, each move is associated with some reward $r \in \{-1, 0, +1\}$. Assume the discount factor $\gamma = 1$ unless otherwise specified.

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

(a) Define the reward r for all states (except state 5 and state 11 whose rewards are specified above) that would cause the optimal policy to return the shortest path to the green target square (no. 5). (3 points)

Solution: Let all rewards be -1. This is because it penalizes those schemes moving many times to get to the green target square. In other words, the fewer times you move, the less negative rewards you receive.

(b) Using r from part (a), find the optimal value function for each square. (5 points)

Solution: With converse thinking, one can start from the green square to obtain the optimal value functions for other squares.

-4	-3	-2	-1	0
5	4	3	2	1
4	-5	2	1	0
-5	-4	-3	-2	-1
-6	-5	-4	-3	-2

(c) Does setting $\gamma = 0.8$ change the optimal policy? Why or why not? (2 points)

Solution: No. Changing γ changes the value function but not the relative order.

Optimization and Machine Learning, Spring 2020
Homework 6
(Due Tuesday, June 2 at 11:59pm (CST))

1. Which of the following sets are convex?

   (a) A *wedge, i.e.*, $\{x \in \mathbb{R}^n | a_1^T x \leq b_1, a_2^T x \leq b_2\}$. (5 points)
   Solution:
   A wedge is an intersection of two halfspaces, so it is convex set.

   (b) The set of points closer to a given points than a given set, *i.e.*,

   $$\{x | ||x - x_0||_2 \leq ||x - y||_2 \text{ for all } y \in S\}$$

   where $S \subseteq \mathbb{R}^n$. (5 points)
   Solution:
   This set is convex because it can be expressed as

   $$\bigcap_{y \in S} \{x | ||x - x_0||_2 \leq ||x - y||_2\}$$

   , which is an intersection of halfspaces

   (c) The set of points closer to one set than another, *i.e.*,

   $$\{x | \mathbf{dist}(x, S) \leq \mathbf{dist}(x, T)\}$$

   where $S, T \subseteq \mathbb{R}^n$, and

   $$\mathbf{dist}(x, S) = \inf\{||x - z||_2 \in S\}.$$

   (5 points)
   Solution:
   In general this set is not convex, as the following example in R shows. With $S = \{-1, 1\}$ and $T = \{0\}$,we have

   $$\{x | \mathbf{dist}(x, S) \leq \mathbf{dist}(x, T)\} = \{x \in R | x \leq -1/2 \text{ or } x \geq 1/2\}$$

   , which is not convex.

   (d) The set $\{x | x + S_2 \subseteq S_1\}$, where $S_1, S_2 \subseteq \mathbb{R}^n$ with S_1 convex. (5 points)
   Solution:
   This set is convex. $x + S_2 \subseteq S_1$ if $x + y \in S_1$, for all $y \in S_2$. Therefore $\{x | x + S_2 \subseteq S_1\} = \bigcap y \in S_2 \{x | x + y \in S_1\} = \bigcap_{y \in S_2} (S_1 - y)$, which is an intersection of convex sets $S_1 - y$

   (e) The set of multiplication

   $$\{x \in \mathbb{R}_+^n | \prod_{i=1}^n x_i \geq 1\}.$$

   (5 points)
   Solution:
   Assume that $\prod_i x_i \geq 1$ and $\prod_i y_i \geq 1$

   $$\prod_i (\theta x_i + (1 - \theta) y_i) \geq \prod_i x_i^{\theta} y_i^{(1-\theta)} \geq 1$$

   So this is convex.

2. Determine whether the following functions are convex, strictly convex, concave, strictly concave, both or neither. (5 points)

(a) $f(x) = e^x - 1$ on $\mathbb{R}$. (5 points)
Solution:

$$f''(x) = e^x > 0$$

So it's strictly convex.

(b) $f(x_1, x_2) = x_1 x_2$ on $\mathbb{R}^2_{++}$. (5 points)
Solution:

$$H(f) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

So it's neither.

(c) $f(x) = \log(\sum_{i=1}^n \exp(x_i))$ on $\mathbb{R}^n$, use the second-order condition. (5 points)
Solution:

$$\nabla^2 f(x) = \frac{1}{\mathbf{1}^T z}\mathbf{diag}(z) - \frac{1}{(\mathbf{1}^T z)^2} z z^T$$

to show $\nabla^2 f(x) \succeq 0$, we must verify $v^T \nabla^2 f(x) v \geq 0$ for all v

$$v^T \nabla^2 f(x) v = \frac{(\sum_k z_k v_k^2)(\sum_k z_k) - (\sum_k v_k z_k)^2}{(\sum_k z_k)^2} \geq 0$$

since $(\sum_k v_k z_k)^2 \leq (\sum_k z_k v_k^2)(\sum_k z_k)$ (from Cauchy-Schwarz inequality)

(d) $f(w) = ||Xw - y||_2^2 + \lambda ||w||_2^2$ for $\lambda > 0$. (5 points)
Solution:

$$\nabla f(x) = 2(Xw - y)^T X + 2\lambda w$$
$$\nabla^2 f(x) = 2X^T X + 2\lambda$$

And it's positive definite matrix. So it's strictly convex

(e) The log-likelihood of a set of points $\{x_1, \cdots, x_n\}$ that are normally distributed with mean μ and finite variance $\sigma > 0$ is given by:

$$f(\mu, \sigma) = n \log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{1}{2\sigma^2}\sum_{i=1}^n (x_i - \mu)^2$$

Show that if we view the log likelihood for fixed σ as a function of the mean, *i.e.,*

$$g(\mu) = n \log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{1}{2\sigma^2}\sum_{i=1}^n (x_i - \mu)^2$$

then g is strictly concave.
Show that if we view the log likelihood for fixed μ as a function of the mean, *i.e.,*

$$h(z) = n \log(\frac{\sqrt{z}}{\sqrt{2\pi}}) - \frac{z}{2}\sum_{i=1}^n (x_i - \mu)^2$$

then h is strictly concave (equivalently, we say f is strictly concave in $z = \frac{1}{\sigma^2}$).
We say $f(x, y)$ with $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^\kappa$ is jointly convex if

$$f(\lambda(x_1, y_1) + (1 - \lambda)(x_2, y_2)) \leq \lambda f((x_1, y_1)) + (1 - \lambda) f((x_2, y_2)).$$

Show that f is not jointly concave in $\mu, \frac{1}{\sigma^2}$. (5 points)
Solution:

$$\nabla g(\mu) = \frac{1}{2\sigma^2}\sum_{i=1}^n 2(x_i - \mu)$$
$$\nabla^2 g(\mu) = \frac{-n}{\sigma^2} < 0$$

So it's strictly concave

$$\nabla^2 h(z) = -\frac{n}{2z^2} < 0$$

So it's strictly concave

$$\nabla^2 f(\mu, \frac{1}{\sigma^2}) = \begin{bmatrix} \frac{-n}{\sigma^2} & \sum_{i=1}^{n}(x_i - \mu) \\ \sum_{i=1}^{n}(x_i - \mu) & -\frac{n\sigma^4}{2} \end{bmatrix}$$

The determinant of the Hessian is given by,

$$det(\nabla^2 f) = \frac{n^2\sigma^2}{2} - (\sum_{i=1}^{n}(x_i - \mu))^2$$

and the trace of the Hessian is given by,

$$tr(\nabla^2 f) = -\frac{n}{\sigma^2} - \frac{n\sigma^4}{2} < 0$$

Note that the trace is the sum of the eigenvalues, and the determinant is the product of the eigenvalues. Since the trace is always negative, if the determinant is negative it must imply that one eigenvalue is positive and another is negative; that is, we have f is neither convex nor concave. It is easy to see that $det(\nabla^2 f)$ can sometimes be negative – for example, if we choose σ^2 to be close to zero and μ away from x_i , the second negative term dominates and make $det(\nabla^2 f) \leq 0$.

3. Consider the problem

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_1 / \left(c^T x + d\right) \\ \text{subject to} & \|x\|_\infty \leq 1 \end{array}$$

where $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ and $d \in \mathbb{R}$. We assume that $d > \|c\|_1$, which implies that $c^{\mathrm{T}} x + d > 0$ for all feasible x.

(a) Show that this is a quasiconvex optimization problem. (5 points)

(b) Show that it is equivalent to the convex optimization problem

$$\begin{array}{ll} \text{minimize} & \|Ay - bt\|_1 \\ \text{subject to} & \|y\|_\infty \leq t \\ & c^T y + dt = 1 \end{array}$$

with variables $y \in \mathbb{R}^n, t \in \mathbb{R}$. (10 points)

Solution:

(a) $f_0(x) \leq \alpha$ if and only if

$$\|Ax - b\|_1 - \alpha(c^T x + d) \leq 0,$$

which is a convex constraint.

(b) Suppose $\|x\|_\infty \leq 1$. We have $c^T x + d > 0$, because $d > \|c\|_1$. Define

$$y = x/(c^T x + d), \quad t = 1/(c^T x + d).$$

Then y and t are feasible in the convex problem with objective value

$$\|Ay - bt\|_1 = \|Ax - b\|_1/(c^T x + d).$$

Conversely, suppose y, t are feasible for the convex problem. We must have $t > 0$, since $t = 0$ would imply $y = 0$, which contradicts $c^T y + dt = 1$. Define

$$x = y/t.$$

Then $\|x\|_\infty \leq 1$, and $c^T x + d = 1/t$, and hence

$$\|Ax - b\|_1/(c^T x + d) = \|Ay - bt\|_1.$$

4. Consider the QCQP

$$\begin{array}{ll} \text{minimize} & (1/2)x^T Px + q^T x + r \\ \text{subject to} & x^T x \leq 1, \end{array}$$

with $P \in \mathbf{S}^n_{++}$. Show that $x^\star = -(P + \lambda I)^{-1} q$ where $\lambda = \max\{0, \bar{\lambda}\}$ and $\bar{\lambda}$ is the largest solution of the nonlinear equation

$$q^{\mathrm{T}}(P + \lambda I)^{-2} q = 1.$$

(15 points)
Solution:
x is optimal if and only if

$$x^T x < 1, \quad Px + q = 0$$

or

$$x^x = 1, \quad Px + q = -\lambda x$$

for some $\lambda \geq 0$. (Geometrically, either x is in the interior of the ball and the gradient vanishes, or x is on the boundary, and the negative gradient is parallel to the outward pointing normal.)

The algorithm goes as follows. First solve $Px = -q$. If the solution has norm less than or equal to one ($\|P^{-1}q\|_2 \leq 1$), it is optimal. Otherwise, from the optimally conditions, x must satisfy $\|x\|_2 = 1$ and $(P + \lambda)x = -q$ for some $\lambda \geq 0$. Define

$$f(\lambda) = \left\|(P + \lambda)^{-1} q\right\|_2^2 = \sum_{i=1}^{n} \frac{q_i^2}{(\lambda + \lambda_i)^2},$$

where $\lambda_i > 0$ are the eigenvalues of P. (Note that $P + \lambda I \succ 0$ for all $\lambda \geq 0$ because $P \succ 0$.) We have $f(0) = \|P^{-1}q\|_2^2 > 1$. Also f monotonically decreases to zero as $\lambda \to \infty$. Therefore the nonlinear equation $f(\lambda) = 1$ has exactly one nonnegative solution $\bar{\lambda}$. Solve for $\bar{\lambda}$. The optimal solution is $x^\star = -(P + \bar{\lambda} I)^{-1} q$.

5. Consider the inequality form LP

$$\begin{array}{ll} \min\limits_{x} & c^T x \\ s.t. & Ax \preceq b, \end{array}$$

with $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$. Let $w \in \mathbb{R}^m_+$. If x is feasible for the LP, *i.e.*, satisfies $Ax \preceq b$, then it also satisfies the inequality

$$w^T Ax \leq w^T b.$$

Geometrically, for any $w \succeq 0$, the halfspace $H_w = \{x \mid w^T Ax \leq w^T b\}$ contains the feasible set for the LP. Therefore if we minimize the objective $c^T x$ over the halfspace Hw we get a lower bound on p^*.

(a) Derive an expression for the minimum value of $c^T x$ over the halfspace H_w (which will depend on the choice of $w \succeq 0$). (5 points)

(b) Formulate the problem of finding the best such bound, by maximizing the lower bound over $w \succeq 0$. (5 points)

(c) Relate the results of (a) and (b) to the Lagrange dual of the LP. (10 points)

Solution:

(a) The optimal value is

$$\inf_{x \in H_w} c^T x = \begin{cases} \lambda w^T b & c = \lambda A^T w \text{ for some } \lambda \leq 0 \\ -\infty & \text{otherwise.} \end{cases}$$

(b) We maximize the lower bound by solving

$$\begin{array}{ll} \max\limits_{\lambda, w} & \lambda w^T b \\ s.t. & c = \lambda A^T w \\ & \lambda \leq 0, \ w \succeq 0 \end{array}$$

with variables λ and w. Note that, as posed, this is not a convex problem.

(c) Defining $z = -\lambda w$, we obtain the equivalent problem

$$\begin{aligned} \max_{z} \quad & -b^T z \\ s.t. \quad & A^T z + c = 0 \\ & z \succeq 0. \end{aligned}$$

This is the dual of the original LP.