University of TEHRAN

Machine Learning- Fall 2019

Final Exam

Department of Bioinformatics, IBB, University of Tehran
January 27, 2020

Part 1: problems 1-5; pages 1-8 **Time limit:** 8:30-10:00

Part 2: problems 6-9; pages 9-14 **Time limit:** 10:15-12:15

Total point: 100

A one-sided cheat sheet is allowed. In addition, you can access to the slides and lecture notes in the second part of the exam (problems 6-9). Good luck!

Your name: ---------- (please *initial all pages!)*

Disclaimer: *some questions are taken or modified from online resources.*

Please leave the below table empty.

PROBLEM	MAXIMUM POINTS	OBTAINED POINTS
1	14	
2	12	
3	8	
4	8	
5	8	
6	5	
7	10	
8	20	
9	15	
TOTAL	100	

Solutions are drafted by Mozhgan Mozaffari Legha (problems 1-5) and Fereshteh Fallah (problem 6-9) and reviewed by Hesam Montazeri (problem 2-6, 8-9) and Kaveh Kavousi (problem 7).

University of TEHRAN

Part 1: problems 1-5

1. Multiple-choice Questions (14 points)

Circle the correct choices. **Note multiple choices or none might be correct.**

i. When the sample size n is extremely large and the number of predictors p is small, a flexible learning method is preferred

a) false b) true

Extremely large sample size (relative to *p)* prevents overfitting to a great extent; hence flexible models do not lead to high variance and are preferred in this setting.

ii. In KNN, K=1 will always give the minimum training error.

a) false b) true

For *K=1*, the predicted response for each test sample is equal to the response of the closest training data point. Since predicted and observed responses are equal for all distinct training samples, the training error is zero in this case. The training error for duplicate data points is not necessarily zero but with the same argument still results in the minimum training error for *K=1*.

iii. In KNN, K=n will always give the minimum generalization error (n: sample size).

a) false b) true

For *K=n*, the predicted response of every point is constant, independent of any feature vector *(x)*. In classification problems, the response is always the most common class available in the training data. The KNN in this case is equivalent to the so-called best constant classifier. This choice often results in large training and test errors due to the high bias of the model.

iv. In statistical learning, the bias always has a bigger contribution to error than the variance.

a) false b) true

False. It depends on model complexity. Variance contributes more to error for complex models while bias contributes more for inflexible models.

v. In random forests, very large number of trees will not result in overfitting.

a) false b) true

Increasing number of trees will reduce variance without a negative impact on bias. The variance reduction is due to the following formula (see class notes for more info; *B* is the number of trees)

$$Var\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

vi. In boosting, very large number of trees will often result in overfitting.

a) false b) true

Boosting approach is a bias-reduction technique. The complexity of boosted trees increases with number of trees. Therefore, an extremely large number of trees can result in overfitting. However, overfitting due to overtraining (or larger number of trees here) often happens at a slower rate in comparison to other non-linear methods (e.g. NNs).

vii. What is the VC dimension for Linear Support Vector Machines in d-dimensional space?
a) 1 b) d c) d+1 d) min (d, n)

viii. Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p}|\beta_j|\le s$$

for a particular value of s. As we increase s from 0, the bias will:
a) Increase initially, and then eventually start decreasing in an inverted U shape.
b) Decrease initially, and then eventually start increasing in a U shape.
c) Steadily increase.
d) steadily decrease.
e) Remain constant.

As we increase s from 0, the β_j coefficients have less constraints and can take values closer to the corresponding least squares estimates; so the model gets more complex with s, hence the bias has to decrease with s.

ix. Repeat the previous question for the test error.
a) Increase initially, and then eventually start decreasing in an inverted U shape.
b) Decrease initially, and then eventually start increasing in a U shape.
c) Steadily increase.
d) Steadily decrease.
e) Remain constant.

The typical U-shape for the test error is expected due to the trade-off between bias and variance. For smaller value of s, the model is simpler and the bias dominates the test error while at the other extreme the variance contributes more for larger values of s.

x. Which of the following models are generative (multiple choices or none can be correct)
a) Logistic regression b) Naïve Bayes c) LDA d) SVM

xi. Bayes error for complex models often is lower than inflexible models.
a) false b) true

Bayes error or irreducible error is due to the fact that features do not completely determine response values and is independent of considered models.

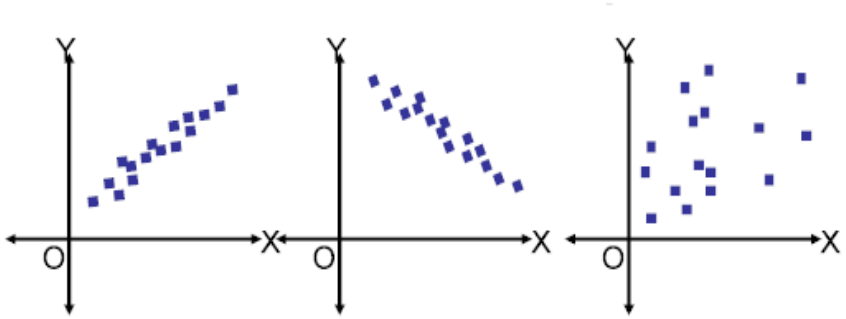
xii. Circle the correct choice(s) about "Type-1" and "Type-2" errors?

a) Type1 is known as false positive and Type2 is known as false negative.
b) Type1 is known as false negative and Type2 is known as false positive.
c) Type1 error occurs when we reject a null hypothesis when it is actually true.

By definition.

xiii. Given below are three scatter plots for two features (Image 1, 2 & 3 from left to right), which choices contain multi-collinear features? Features in

a) Left image b) Middle image c) Right image



"Multicollinearity refers to a situation in which two or more explanatory variables in a multiple regression model are highly linearly related" (wiki). X and Y are highly correlated in left and middle images.

xiv. In a binary classification problem, circle all choices that are correct according to the below confusion matrix:

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

a) Accuracy is ~0.91 b) Misclassification rate is ~ 0.91

c) False positive rate is ~0.95

d) True positive rate is ~0.95

$$Accuracy = \frac{100 + 50}{50 + 10 + 5 + 100} \approx 0.91$$
$$\text{Misclassification rate } = \frac{15}{165} \approx 0.09$$

$$False\ positive\ rate = \frac{FP}{N} = \frac{10}{10 + 50} \approx 0.167$$
$$True\ \ positive\ rate = \frac{TP}{P} = \frac{100}{100 + 5} \approx 0.95$$

2. Short Questions (12 points)

a) What is the difference between classification and regression models?

The response variable is continuous for a regression problem while is categorical for a classification task.

b) Define Bias and Variance of an estimator. What is the bias-variance decomposition?

$bias(\hat{\theta}) = E(\hat{\theta}) - \theta$

$Var(\hat{\theta}) = \mathrm{E}(\hat{\theta}^2) - E^2(\hat{\theta})$

By some algebra it is easy to show reducible error can be decomposed as follows:

$mse(\hat{\theta}) = E(\hat{\theta} - \theta)^2 = bias^2(\hat{\theta}) + Var(\hat{\theta})$

c) What is the difference between stochastic and batch gradient descent algorithms?

Batch gradient descent computes the gradient using the whole dataset while stochastic gradient descent estimates the gradient using a single sample.

d) Explain random forest algorithm. Why is it a variance reduction algorithm?

Random forest is a variation of the bagging technique that attempts to further reduce variance by using de-correlated trees through random feature selection at each split (hence smaller ρ in the following formula and consequently lower variance).

$$Var\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

Please refer to the class notes for more details.

e) Explain how maximum likelihood estimation is different for discriminative and generative models (Hint: you need to explain in terms of likelihood function)

The likelihood of generative models is expressed in terms of the joint probability distribution P(X,Y). The estimated model can be used to generate new realizations of (X, Y) using the estimated joint distribution. Application of Bayes' rule is required for computing the conditional probability P(Y|X). On the other hand, the likelihood of discriminative models is written in terms of conditional probabilities P(Y|X).

f) Explain the difference between parametric and non-parametric models. Give an example for each.

Suppose the true but unknown function *f* encodes the relationship between X and Y. A parametric model assumes f has a specific functional form with a finite set of parameters (e.g. a linear model). In parametric models, we make some assumptions (e.g. linearity) on the functional form of *f*. The estimation of f is then reduced to the estimation of model parameters. It is noteworthy that the number of parameters does not vary with sample size.

"Non-parametric methods do not make explicit assumptions about the functional form of *f*. Instead they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly" [from ISL]. In parametric models, the degree of freedom can vary with sample size. KNN is a non-parametric model.

3. Support vector machines (8 points)

In the below Figure, there are different SVMs with different shapes/patterns of decision boundaries. The training data is labeled as $y[i] \in \{-1, 1\}$, represented as the shape of circles and squares respectively. Support vectors are shaded. Match the scenarios described below to one of the four plots. Each scenario should be matched to a unique plot. Explain in two sentences why it is the case for each scenario.

a) A soft-margin linear SVM with $C = 0.02$.
b) A soft-margin linear SVM with $C = 20$.

Since the linear SVM has a linear decision boundary, Fig. 3 or 4 corresponds to scenarios *a* and *b*. Through basic analysis of fitness and complexity, we know the larger *C* in SVM results in more complex models, hence narrower margin. Hence, scenarios *a* and *b* correspond to figures 4 and 3, respectively.

Another insight is to compare the number of support vectors for narrow and wide margins which can assist to correspond the scenarios to the figures. It is noteworthy when the margin is wide "many observations violate the margin, and so there are many support vectors. In this case, many observations are involved in determining the hyperplane.". On the other hand, when the margin is narrow "then there will be fewer support vectors and hence the resulting classifier will have low bias but high variance." (From ISL).

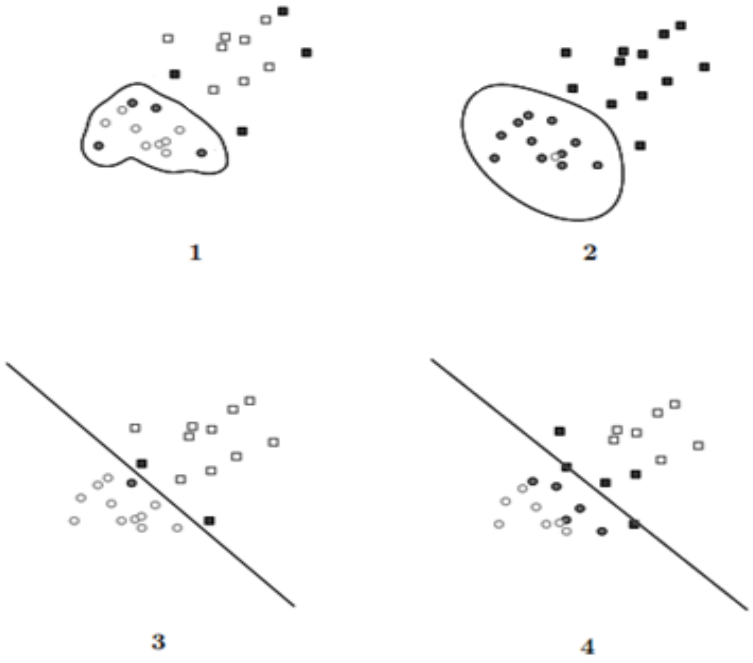
c) A hard-margin kernel SVM with $K(u,v) = \exp(-5\|u-v\|^2)$

d) A hard-margin kernel SVM with $K(u,v) = \exp(-\frac{1}{5}\|u-v\|^2)$

$K(u,v) = \exp(-\delta\|u-v\|^2)$ defines the similarity of two data points. Larger values of δ make *u* and *v* less similar. The SVM function using the kernel trick has the following form

$$f(x) = \beta_0 + \sum_{i \epsilon S} \alpha_i K(x, x_i)$$

For large values of δ, f(x) will be dominated by a small set of data points x_i in the restricted neighborhood of *x*, hence we expect a more complex model in this case. The argument presented here is similar to the analysis of parameter *K* in the KNN model (in this case larger δ corresponds to the lower K). So scenarios c and *d* correspond to figures 1 and 2, respectively.



4. Model selection (8 points)

For each of the following dataset, explain which learning model or models do you use? (N: number of observations; p: number of feature)

a) Prediction task: $N = 20,\ p = 10^6$
Since $p \gg N$, using a complex model can result in overfitting. Regularized regression/classification models (ridge or lasso) are among good choices here.

b) Knowledge discovery task: $N = 20,\ p = 10^6$

We need an interpretable model that can avoid overfitting. Lasso is one of the few choices for this setting because it sheds light on important predictors through its variable selection operator. It is noteworthy to acknowledge it is an extremely challenging task and depending on the input data estimating any (accurate) model might not be feasible for this task.

c) Prediction task: $N = 10^9,\ p = 10^3$

Because $N \gg p$, overfitting is of less concern; hence we may use complex model such as neural networks.

d) Knowledge discovery task: $N = 10^9,\ p = 10^3$

In comparison to the task at part *c*, the interpretability is of more importance. Using the linear regression models or lasso models with inclusion of all pairwise interaction terms seem reasonable.

5. Stochastic gradient ascent rule for Poisson regression (8 points)

Poisson regression is a generalized linear model where the response variable is assumed to have a Poisson distribution.

Recall a discrete random variable Y is said to have a Poisson distribution with parameter λ > 0, if, for y = 0, 1, 2, ..., the probability mass function of Y is given by $\Pr(\mathrm{Y} = \mathrm{y}) = \frac{\lambda^y e^{-\lambda}}{y!}$. In addition, a distribution is from an exponential family if it can be expressed in the form $P_\theta(y) = \frac{1}{Z(\theta)} h(y)\exp(t(\theta)^T S(y))$ where θ is the parameter set of P and Z, h, t, and S functions are some known functions. In this problem you need to

a) Write the Poisson distribution in an exponential family form.

The Poisson PMF is

$$f(y) = \Pr(\mathrm{Y} = \mathrm{y}) = \frac{\lambda^y e^{-\lambda}}{y!}$$

defined for $y \geq 0$. We can rewrite the PMF as

$$f(y) = \frac{1}{y!} e^{-\lambda} \exp(\log \lambda^y) = \frac{1}{y!} e^{-\lambda} \exp(\mathrm{y} \log \lambda)$$

Hence using the below functions we have $f(y)$ in the exponential family form on the same support as $f(y)$

$$h(y) = \frac{1}{y!}\ , Z(\lambda) = e^{\lambda}\ , S(y) = y\ , t(\lambda) = \log \lambda$$

b) Derive the stochastic gradient ascent rule for Poisson regression.

We first derive the response function by considering

$$t(\lambda) = \beta^T x \ \Rightarrow\ log\lambda = \beta^T x \Rightarrow \lambda = e^{\beta^T x} \qquad (1)$$

and since Y is a Poisson random variable we have

$$f(x) = E(Y|x, \beta) = \lambda \qquad (2)$$

From (1) and (2), it is evident $f(x) = e^{\beta^T x}$

Since the goal is to derive the stochastic gradient ascent rule, we write down the likelihood function for only a single observation $(y[i], x[i])$:

$$L(\beta) = \mathrm{P}(y[i] \mid x[i], \beta) = \frac{\lambda^{y[i]} e^{-\lambda}}{y[i]!}$$

where $\lambda = \exp(\beta^{\mathrm{T}} x[i])$. The log-likelihood $l(\beta)$ is then

$$l(\beta) = y[i]\beta^T x[i] - \exp(\beta^T x[i])$$

and its gradient is

$$\frac{\partial l(\beta)}{\partial \beta} = y[i]x[i] - x[i]\exp(\beta^T x[i])$$

hence the updating formula for i^{th} observation is

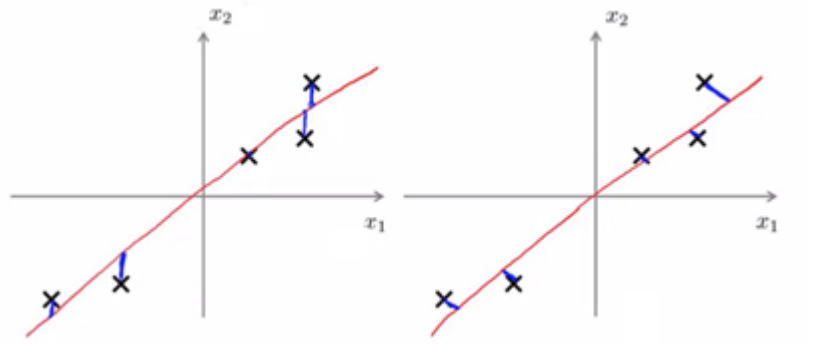
$$\beta_{new} := \beta + \alpha(y[i]x[i] - x[i]\exp(\beta^T x[i]))$$

where α is the learning rate.

Part 2: problems 6-9

6. PCA and linear regression (5 points)

The first principle component of the PCA model and the linear regression model are drawn in the below figures. Explain which is which and why.



The right figure indicates an orthogonal projection of data on a line with the greatest variance of data on the projected line; hence it corresponds to the PCA.

The left figure corresponds to the regression model. The cross symbols indicate observed data points. For each data point, $x_1[i]$ and $x_2[i]$ are predictor and response variables, respectively. For each $x_1[i]$, the linear function provides the predicted response ($\hat{f}(x_1[i])$). The difference $\hat{f}(x_1[i]) - x_2[i]$ corresponds to the residual for *i*th observation.

7. Clustering (10 points)

The pseudo code of *BSAS (Basic Sequential Algorithmic Scheme)* and *MBSAS (Modified BSAS)* clustering algorithms are presented here:

BSAS	***MBSAS***
• $m = 1$ • $C_m = \{\boldsymbol{x}_1\}$ • For $i = 2$ to N —Find C_k: $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$ —If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$ then * $m = m + 1$ * $C_m = \{\boldsymbol{x}_i\}$ —Else * $C_k = C_k \cup \{\boldsymbol{x}_i\}$ * update representative. —End {if} • End {For}	*Cluster Determination* • $m = 1$ • $C_m = \{\boldsymbol{x}_1\}$ • For $i = 2$ to N —Find C_k: $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$ —If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$ then * $m = m + 1$ * $C_m = \{\boldsymbol{x}_i\}$ —End {if} • End {For} *Pattern Classification* • For $i = 1$ to N —If $\boldsymbol{x}_i$ has not been assigned to a cluster, then * Find C_k: $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$ * $C_k = C_k \cup \{\boldsymbol{x}_i\}$ * update representative. —End {if} • End {For}

The parameters of both algorithms: Θ: Threshold of dissimilarity, N: Number of samples, q: The maximum allowable number of clusters , m: Current number of clusters

$d(x, C) = d(x, m_C)$: d is the distance between sample x and cluster C where m_C is the representative of cluster C. Consider mean as representative. Update m_{C_k} in each step as follows (n is the cardinality of C_k after the assignment of x to it): $$\boldsymbol{m}_{C_k}^{new} = \frac{(n-1)\boldsymbol{m}_{C_k}^{old} + \boldsymbol{x}}{n}$$

Now, consider the following two-dimensional vectors:

$x_1 = [1,1]^T, x_2 = [1,2]^T, x_3 = [2,2]^T, x_4 = [2,3]^T, x_5 = [3,3]^T, x_6 = [3,4]^T, x_7 = [4,4]^T,$

$x_8 = [4,5]^T, x_9 = [5,5]^T, x_{10} = [5,6]^T, x_{11} = [-4,5]^T, x_{12} = [-3,5]^T, x_{13} = [-4,4]^T, x_{14} = [-3,4]^T$

a) Run ***MBSAS*** when the vectors are presented in the given order. Use the Euclidean distance between two vectors and take $\Theta = \sqrt{2}$. Write the final results below and show your work. (**Hint:** first plot the points and then follow the steps of the algorithm).

MBSAS: C_1 = { }, C_2 = { }, C_3 = { }, C_4 = { } and C_5 = { }

MBSAS algorithm consists of two phases. The clusters are determined in the first phase via assignment of some vectors of *X* to them. The unassigned vectors are assigned to the appropriate clusters in the second phase.

The first phase:

A cluster is formed during the first pass, every time the distance of a vector from the already formed cluster is larger than $\sqrt{2}$.

The result of the first phase:

$$C_1 = x_1 \ , \ m_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \ C_2 = x_4 \ , \ m_4 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \ C_3 = x_7 \ , \ m_7 = \begin{pmatrix} 4 \\ 4 \end{pmatrix},$$

$$C_4 = x_{10} \ , \ m_{10} = \begin{pmatrix} 5 \\ 6 \end{pmatrix}, \ C_5 = x_{11} \ , \ m_{11} = \begin{pmatrix} -4 \\ 5 \end{pmatrix}$$

The second phase:

During the second pass, unassigned vectors to the current cluster are assigned based on minimum distance criteria.

$$\xrightarrow{\text{Decision on } x_3} d(C_1, x_2) < d(C_i, x_2) \ , i = 2,3,4,5 \Rightarrow C_1 = \{x_1, x_2\} \ , \ m_1 = \begin{pmatrix} 1 \\ 3/2 \end{pmatrix}$$

$$\xrightarrow{\text{Decision on } x_3} \begin{cases} d(C_1, x_3) = 1/1 \\ d(C_2, x_3) = 1 \end{cases} \Rightarrow \begin{cases} C_1 = \{x_1, x_2\} \ , \ m_1 = \begin{pmatrix} 1 \\ 3/2 \end{pmatrix} \\ C_2 = \{x_3, x_4\} \ , \ m_2 = \begin{pmatrix} 2 \\ 5/2 \end{pmatrix} \end{cases}$$

$$\xrightarrow{\text{Decision on } x_5}\begin{cases} d(C_2,x_5)=\sqrt{\begin{pmatrix}3\\3\end{pmatrix}-\begin{pmatrix}2\\5/2\end{pmatrix}=\begin{pmatrix}1\\1/2\end{pmatrix}}=1/1 \\ d(C_3,x_5)=\sqrt{\begin{pmatrix}3\\3\end{pmatrix}-\begin{pmatrix}4\\4\end{pmatrix}=\begin{pmatrix}1\\1\end{pmatrix}}=\sqrt{2} \end{cases} \Rightarrow C_2=\{x_3,x_4,x_5\}\ ,\ m_2=\begin{pmatrix}7/3\\8/3\end{pmatrix}$$

$$\xrightarrow{\text{Decision on } x_6}\begin{cases} d(C_2,x_6)=1.25 \\ d(C_3,x_6)=1 \end{cases} \Rightarrow C_3=\{x_6,x_7\}\ ,\ m_3=\begin{pmatrix}7/2\\4\end{pmatrix}$$

$$\xrightarrow{\text{Decision on } x_8}\begin{cases} d(C_3,x_8)=\sqrt{\begin{pmatrix}4\\5\end{pmatrix}-\begin{pmatrix}7/2\\4\end{pmatrix}}\simeq 1/1 \\ d(C_4,x_8)=\sqrt{\begin{pmatrix}4\\5\end{pmatrix}-\begin{pmatrix}5\\6\end{pmatrix}}\simeq 1/4 \end{cases} \Rightarrow C_3=\{x_6,x_7,x_8\}\ ,\ m_3=\begin{pmatrix}11/3\\13/3\end{pmatrix}$$

$$\xrightarrow{\text{Decision on } x_9}\begin{cases} d(C_3,x_9)=\sqrt{\begin{pmatrix}5\\5\end{pmatrix}-\begin{pmatrix}11/3\\13/3\end{pmatrix}}\simeq 1/8 \\ d(C_4,x_9)=\sqrt{\begin{pmatrix}5\\5\end{pmatrix}-\begin{pmatrix}5\\6\end{pmatrix}}\simeq 1 \end{cases} \Rightarrow C_4=\{x_9,x_{10}\}\ ,\ m_4=\begin{pmatrix}5\\11/2\end{pmatrix}$$

$$\xrightarrow{\text{Decision on } x_{12}}\begin{cases} d(C_4,x_{12})=\sqrt{\begin{pmatrix}-3\\5\end{pmatrix}-\begin{pmatrix}5\\6\end{pmatrix}}\simeq 8/02 \\ d(C_5,x_{12})=\sqrt{\begin{pmatrix}-3\\5\end{pmatrix}-\begin{pmatrix}-4\\5\end{pmatrix}}\simeq 1 \end{cases} \Rightarrow C_5=\{x_{11},x_{12}\}\ ,\ m_5=\begin{pmatrix}-7/2\\5\end{pmatrix}$$

$$\xrightarrow{\text{Decision on } x_{13}} d(C_5,x_{13})=\sqrt{\begin{pmatrix}-4\\4\end{pmatrix}-\begin{pmatrix}-3/5\\5\end{pmatrix}}\simeq 1/1<d(C_i,x_{13})\ ,\ i=1,2,3,4\Rightarrow C_5=\{x_{11},x_{12},x_{13}\}\ ,\ m_5=\begin{pmatrix}-11/3\\14/3\end{pmatrix}$$

$$\xrightarrow{\text{Decision on } x_{14}} d(C_5,x_{14})\simeq 0/94<d(C_i,x_{14})\ ,\ i=1,2,3,4\Rightarrow C_5=\{x_{11},x_{12},x_{13},x_{14}\}\ ,\ m_5=\begin{pmatrix}-14/4\\18/4\end{pmatrix}$$

The final result:

$$C_1=\{x_1,x_2\}\quad ,\quad C_2=\{x_3,x_4,x_5\}\quad ,\quad C_3=\{x_6,x_7,x_8\}$$
$$C_4=\{x_9,x_{10}\}\quad ,\quad C_5=\{x_{11},x_{12},x_{13},x_{14}\}$$

b) Discuss sensitivity of the *BSAS* and *MBSAS* to the ordering of presentation of the vectors to the algorithm.

Both algorithms are sensitive to the order of presentation of vectors. In other words, if the order of input changes, it is likely that the result of the algorithms change too.

8. Boosting (20 points)

In this question, you need to investigate how Adaboost works for the XOR problem. The Adaboost algorithm is as follows

Algorithm 10.1 *AdaBoost.M1.*

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

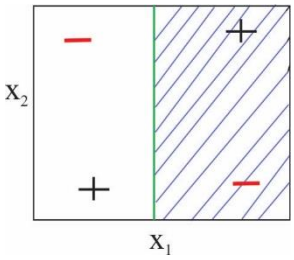
<u>For the parts *a* and *b*, assume the weak learner is a decision stump (decision tree of with maximum two leaves).</u>

a) Using the below dataset, draw the decision boundary learned by G_1 in the first iteration. What is α_1 ? *<u>(mark regions as positive/negative assuming that ties are broken arbitrarily.)</u>*

In the first iteration, we have

$$m = 1; \quad w_i = \frac{1}{4} \quad for \quad i = 1, \ldots, 4$$

then



$$err = \frac{\sum_{i=1}^{4} 1/4 \times I(y_i \neq G_1(x_i))}{\sum_{i=1}^{4} 1/4} = \frac{\frac{1}{4}\times 0 + \frac{1}{4}\times 1 + \frac{1}{4}\times 0 + \frac{1}{4}\times 1}{1} = \frac{1}{2}$$

$$\alpha_1 = \log(\frac{1-err}{err}) = \log(\frac{1-1/2}{1/2}) = 0$$

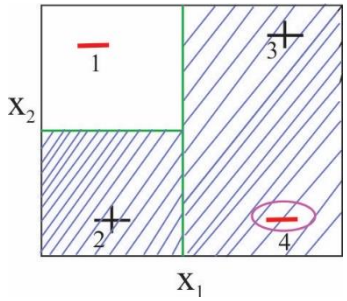
The estimated decision stump is a random classifier because the $error = 0.5$. Aggregation of random classifiers does not improve the accuracy of the boosted tree hence we have $\alpha_1 = 0$.

b) What is the minimum achievable training error by Adaboost for this example? Why do you think so?

According to the previous part, estimated decision stump is a random classifier and each has a zero weight in the boosted classifier. Hence, the boosted classifier always classifies input sample as positive which yields the training error of 0.5 (see the step 3 of the algorithm).

For the remaining parts, assume the weak learner is a decision tree of with maximum three leaves.

c) Draw the decision boundary learned by G_1. Compute α_1 and weights. Circle misclassified points by G_1.



For the new weak learner, we start with

$$m = 1; \quad w_i = \frac{1}{4} \quad for \;\; i = 1, \ldots, 4$$

Then we obtain

$$err = \frac{\sum_{i=1}^{4} 1/4 \times I(y_i \neq G_1(x_i))}{\sum_{i=1}^{4} 1/4} = \frac{\frac{1}{4}\times 0 + \frac{1}{4}\times 0 + \frac{1}{4}\times 0 + \frac{1}{4}\times 1}{1} = \frac{1}{4}$$

$$\alpha_1 = \log(\frac{1-err}{err}) = \log(\frac{1-1/4}{1/4}) = \log 3$$

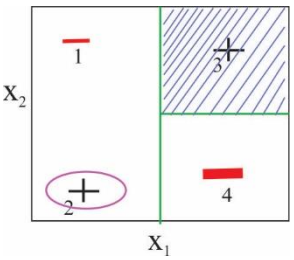
The new weights are then

$$w_1 = w_2 = w_3 = \frac{1}{4}, \; w_4 = \frac{3}{4}$$

the shaded areas in this and subsequent figures denote the positive regions as determined by the current weak learner (G_1 in this part).

d) Draw the decision boundary learned by G_2. Compute α_2 and weights. Circle misclassified points by G_2.

In the second iteration ($m = 2$), we first update



$$err = \frac{\sum_{i=1}^{4} w_i \times I(y_i \neq G_2(x_i))}{\sum_{i=1}^{4} w_i} = \frac{\frac{1}{4}\times 0+\frac{1}{4}\times 1+\frac{1}{4}\times 0+\frac{3}{4}\times 0}{\frac{6}{4}} = \frac{1}{6}$$

$$\alpha_2 = \log(\frac{1-err}{err}) = \log(\frac{1-1/6}{1/6}) = \log 5$$

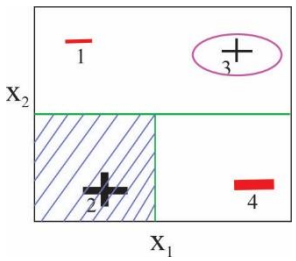
The new weights are then

$$w_1 = w_3 = \frac{1}{4}, w_2 = \frac{5}{4}, w_4 = \frac{3}{4}$$

e) Draw the decision boundary learned by G_3. Compute α_3 and weights. Circle misclassified points by G_3.

$$err = \frac{\sum_{i=1}^{4} w_i \times I(y_i \neq G_3(x_i))}{\sum_{i=1}^{4} w_i} = \frac{\frac{1}{4}\times 0+\frac{5}{4}\times 0+\frac{1}{4}\times 1+\frac{3}{4}\times 0}{\frac{10}{4}} = \frac{1}{10}$$

$$\alpha_3 = \log(\frac{1-err}{err}) = \log(\frac{1-1/10}{1/10}) = \log 9$$



The new weights are then

$$w_1 = \frac{1}{4}, w_2 = \frac{5}{4}, w_3 = \frac{9}{4}, w_4 = \frac{3}{4}$$

f) Indicate how $G = sgn(\alpha_1 G_1 + \alpha_2 G_2 + \alpha_3 G_3)$ classify each of the four points?

$$\text{prediction point } 1 = \text{sgn}(\alpha_1 G_1 + \alpha_2 G_2 + \alpha_3 G_3) = \text{sgn}(\log 3\times -1 + \log 5\times -1 + \log 9\times -1) = \text{sgn}(-4.90) = -1$$
$$\text{prediction point } 2 = \text{sgn}(\alpha_1 G_1 + \alpha_2 G_2 + \alpha_3 G_3) = \text{sgn}(\log 3\times +1 + \log 5\times +1 + \log 9\times -1) = \text{sgn}(0.51) = +1$$
$$\text{prediction point } 3 = \text{sgn}(\alpha_1 G_1 + \alpha_2 G_2 + \alpha_3 G_3) = \text{sgn}(\log 3\times +1 + \log 5\times -1 + \log 9\times +1) = \text{sgn}(1.68) = +1$$
$$\text{prediction point } 4 = \text{sgn}(\alpha_1 G_1 + \alpha_2 G_2 + \alpha_3 G_3) = \text{sgn}(\log 3\times +1 + \log 5\times -1 + \log 9\times -1) = \text{sgn}(-2.70) = -1$$

h) What is the minimum training error achievable by boosting for this example?

According to the part f, Adaboost already correctly classifies all points in the third iteration so the minimum achievable training error is zero.

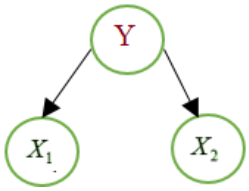
9. Bayesian inference for Naïve Bayes classifier (15 points)

The goal is to provide a Bayesian inference for the Naïve Bayes model with binary features and response variable. Assume the input data is D=$(x[i], y[i])$ $i = 1, \dots, N$ where N is the number of observations and $x[i] = \left[x_1[i], \dots, x_p[i]\right]^T$ (p: number of features).

a) Write down the model, its parameters, and likelihood function for p=2.

Model:

$$\begin{aligned} Y &\sim \text{Bernoulli}(\theta_Y) \\ X_j \mid Y=1 &\sim \text{Bernoulli}(\theta_{X_j|Y=1}) \\ X_j \mid Y=0 &\sim \text{Bernoulli}(\theta_{X_j|Y=0}) \end{aligned} \quad j=1,2$$



Parameters:

$$\theta = (\theta_Y, \theta_{X_1|Y=1}, \theta_{X_1|Y=0}, \theta_{X_2|Y=1}, \theta_{X_2|Y=0})$$

Likelihood:

Since the Naïve Bayes model is a generative model, we write the likelihood based on the joint probability:

$$L(\theta) = P(D \mid \theta) = \prod_{i=1}^{n} P(x[i], y[i]) = \prod_{i=1}^{n} P(x[i] \mid y[i]) P(y[i])$$

$$\xrightarrow{NB\ assumption} \prod_{i=1}^{n} P(x_1[i] \mid y[i]=1) P(x_2[i] \mid y[i]=1) P(y[i])$$

The conditional probabilities are given by

$$\begin{aligned} P(x_1[i] \mid y[i]=1) &= (\theta_{X_1|Y=1})^{x_1[i]} (1-\theta_{X_1|Y=1})^{1-x_1[i]} \\ P(x_2[i] \mid y[i]=1) &= (\theta_{X_2|Y=1})^{x_2[i]} (1-\theta_{X_2|Y=1})^{1-x_2[i]} \\ P(y[i]) &= (\theta_Y)^{y[i]} (1-\theta_Y)^{1-y[i]} \end{aligned}$$

The likelihood function is then simplified as

$$L(\theta) = \prod_{i=1}^{n} (\theta_{X_1|Y=1})^{x_1[i]} (1-\theta_{X_1|Y=1})^{1-x_1[i]} (\theta_{X_2|Y=1})^{x_2[i]} (1-\theta_{X_2|Y=1})^{1-x_2[i]} (\theta_Y)^{y[i]} (1-\theta_Y)^{1-y[i]}$$

b) For Bayesian inference, choose the beta prior for parameters (you may choose the same prior for all parameters). Write down the posterior distribution and the MAP estimate for parameters. Necessary formula for Beta distribution are given at the end of this question.

We pick the same beta prior for all parameters

$$\theta_{X_1|Y=0}, \theta_{X_1|Y=1}, \theta_{X_0|Y=0}, \theta_{X_0|Y=1}, \theta_Y \sim Beta(\alpha, \beta)$$

Since beta is a conjugate prior for the Bernoulli likelihood function, the posterior distributions are beta with new hyperparameters:

$$\xrightarrow{conjugate} \theta_{X_1|Y=1} \mid D \sim Beta(\overbrace{\alpha + M[X_1 = 1, Y = 1]}^{\alpha_1}, \overbrace{\beta + M[X_1 = 0, Y = 1]}^{\beta_1})$$

$$\xrightarrow{conjugate} \theta_{X_1|Y=0} \mid D \sim Beta(\overbrace{\alpha + M[X_1 = 1, Y = 0]}^{\alpha_2}, \overbrace{\beta + M[X_1 = 1, Y = 0]}^{\beta_2})$$

$$\xrightarrow{conjugate} \theta_{X_2|Y=1} \mid D \sim Beta(\overbrace{\alpha + M[X_2 = 1, Y = 1]}^{\alpha_3}, \overbrace{\beta + M[X_2 = 0, Y = 1]}^{\beta_3})$$

$$\xrightarrow{conjugate} \theta_{X_1|Y=1} \mid D \sim Beta(\overbrace{\alpha + M[X_2 = 1, Y = 0]}^{\alpha_4}, \overbrace{\beta + M[X_2 = 0, Y = 0]}^{\beta_4})$$

$$\xrightarrow{conjugate} \theta_Y \mid D \sim Beta(\overbrace{\alpha + M[Y = 1]}^{\alpha_5}, \overbrace{\beta + M[Y = 0]}^{\beta_5})$$

where $M[.,.]$ denotes number of observations with given criteria. To obtain the MAP estimates, we use the formula for the mode of the beta distribution.

$$\hat{\theta}^{MAP}_{X_1|Y=1} = \frac{\alpha_1 - 1}{\alpha_1 + \beta_1 - 2}, \hat{\theta}^{MAP}_{X_1|Y=0} = \frac{\alpha_2 - 1}{\alpha_2 + \beta_2 - 2}, \hat{\theta}^{MAP}_{X_2|Y=1} = \frac{\alpha_3 - 1}{\alpha_3 + \beta_3 - 2}$$

$$\hat{\theta}^{MAP}_{X_2|Y=0} = \frac{\alpha_4 - 1}{\alpha_4 + \beta_4 - 2}, \hat{\theta}^{MAP}_{Y} = \frac{\alpha_5 - 1}{\alpha_5 + \beta_5 - 2}$$

c) Despite the fact that the closed-form expression for posterior distribution is known, for educational reasons, your task is to develop an MCMC algorithm for estimating the posterior distribution.

The MCMC algorithm:

Start from an initial $\theta^{(0)} = (\theta_{X_1|Y=1}{}^{(0)}, \theta_{X_1|Y=0}{}^{(0)}, \theta_{X_2|Y=1}{}^{(0)}, \theta_{X_2|Y=0}{}^{(0)}, \theta_Y{}^{(0)})$

For $k = 0, \cdots, n-1$

Sample θ' from $Q(\theta' \mid \theta^{(k)})$

with probability $\min(\frac{L(\theta')}{L(\theta^{(k)})} \cdot \frac{Q(\theta^{(k)} \mid \theta')}{Q(\theta' \mid \theta^{(k)})}, 1)$

set $\theta^{(k+1)} = \theta'$

otherwise $\theta^{(k+1)} = \theta^{(k)}$

output $\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(n)}$

Since Bernoulli parameters are bounded between 0 and 1, there are several possibilities for the proposal distribution *Q*:

1) Next parameter value is independent of the current value.

$$\theta_j^{(k+1)} \mid \theta_j^k \sim Unif(0,1) \qquad for\ j = 1,\ldots,5$$

2) Another possibility is to use the following beta distribution

$$\theta_j^{(k+1)} \mid \theta_j^k \sim Beta(c\theta_j^k, c(1-\theta_j^k)) \qquad for\ j = 1,\ldots,5$$

The expected of the next value, $\theta_j^{(k+1)}$, is the current value θ_j^k. The parameter c determines to what extent the next value, $\theta_j^{(k+1)}$, will be concentrated around the current value, θ_j^k.

Notation	Beta(α, β)
Parameters	$\alpha > 0$ shape (real) $\beta > 0$ shape (real)
Support	$x \in [0,1]$ or $x \in (0,1)$
PDF	$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$ where $\mathrm{B}(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and Γ is the Gamma function.
CDF	$I_x(\alpha,\beta)$ (the regularised incomplete beta function)
Mean	$\mathrm{E}[X] = \frac{\alpha}{\alpha+\beta}$ $\mathrm{E}[\ln X] = \psi(\alpha) - \psi(\alpha+\beta)$ $\mathrm{E}[X \ln X] = \frac{\alpha}{\alpha+\beta}\left[\psi(\alpha+1) - \psi(\alpha+\beta+1)\right]$ (see digamma function and see section: Geometric mean)

Median	$I^{[-1]}_{\frac{1}{2}}(\alpha,\beta)$ (in general) $\approx \frac{\alpha-\frac{1}{3}}{\alpha+\beta-\frac{2}{3}}$ for $\alpha,\beta > 1$
Mode	$\frac{\alpha-1}{\alpha+\beta-2}$ for $\alpha, \beta > 1$ any value in $(0,1)$ for $\alpha, \beta = 1$ {0, 1} (bimodal) for $\alpha, \beta < 1$ 0 for $\alpha \le 1, \beta > 1$ 1 for $\alpha > 1, \beta \le 1$
Variance	$\mathrm{var}[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$ $\mathrm{var}[\ln X] = \psi_1(\alpha) - \psi_1(\alpha+\beta)$ (see trigamma function and see section: Geometric variance)