HOMEWORK 3

DEEPAN DAS
ddas27

Instructions: Although this is a programming homework, you only need to hand in a pdf answer file. There is no need to submit the latex source or any code. You can choose any programming language, as long as you implement the algorithm from scratch.
Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Please check Piazza for updates about the homework.

1 A Simplified 1NN Classifier

You are to implement a 1-nearest-neighbor learner for classification. To simplify your work, your program can assume that

- each item has d continuous features $\mathbf{x} \in \mathbb{R}^d$
- binary classification and the class label is encoded as $y \in \{0, 1\}$
- data files are in plaintext with one labeled item per line, separated by whitespace:

$$\begin{array}{cccc} x_{11} & \dots & x_{1d} & y_1 \\ & \dots & & \\ x_{n1} & \dots & x_{nd} & y_n \end{array}$$

Your program should implement a 1NN classifier:

- Use Mahalanobis distance d_A parametrized by a positive semidefinite (PSD) diagonal matrix A. For $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$,
$$d_A(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_A = \sqrt{(\mathbf{x} - x')^\top A(\mathbf{x} - x')}.$$
We will specify A in the questions below. (Hint: d is dimension while d_A with a subscript is distance)
- If multiple training points are the equidistant nearest neighbors of a test point, you may use any one of those training points to predict the label.
- You do not have to implement kd-tree.

2 Questions

1. (5 pts) What is the mathematical condition on the diagonal elements for a diagonal matrix A to be PSD?

   A matrix M is PSD for the following condition:

   $$A \quad PSD \Leftrightarrow x^T Ax \geq 0 \quad \forall x \in \mathbb{R}^n$$

   Thus, if $A \in \mathbb{R}^{dxd}$ and its diagonal elements are $a_1, a_2, ..., a_d$, and let $x \in \mathbb{R}^d$ be defined as $[x_1, x_2, ..., x_d]$. Thus, $x^T Ax \geq 0$ evaluates to:

   $$x_1^2 a_1 + x_2^2 a_2 + ... + x_d^2 a_d \geq 0$$

   Thus, we see, that all diagonal elements $\{a_1, a_2, ..., a_d\}$ should be greater than equal to zero. Or, for every a_i in $\{a_1, a_2, ..., a_d\}$, $a_i \geq 0$

2. (5 pts) Given a training data set D, how do we preprocess it to make each feature dimension mean 0 and variance 1? (Hint: give the formula for $\hat{\mu}_j, \hat{\sigma}_j$ for each dimension j, and explain how to use them to normalize the data. You may use either the $\frac{1}{n}$ or $\frac{1}{n-1}$ version of sample variance. You may assume the sample variances are non-zero.)

   Let x_j be a feature in the dataset, where $x_j = [x_{1j}, x_{2j}, ..., x_{nj}]$. Then, the following operation can be applied to all the x_{ij} in x_j so as to make x_j a feature $\tilde{x}_j$ with zero mean and unit variance.

$$\tilde{x}_{i,j} = \frac{x_{i,j} - \hat{\mu}_j}{\hat{\sigma}_j} \quad \forall x_{ij} \in [x_{1j}, x_{2j}, ..., x_{nj}]$$

   Where, $\hat{\mu}_j$ refers to the mean across that j-th feature, and $\hat{\sigma}_j$ refers to the standard deviation

$$\hat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

$$\hat{\sigma}_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \hat{\mu}_j)^2}$$

3. (5 pts) Let $\tilde{\mathbf{x}}$ be the preprocessed data. Give the formula for the Euclidean distance between $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}'$.

   The Euclidian distance can be expressed as :

$$dist_{Euclidian}(\tilde{x}, \tilde{x}') = \sqrt{\sum_{i=1}^{d}(\tilde{x}_i - \tilde{x}'_i)^2} = \sqrt{(\tilde{x} - \tilde{x}')^\top I(\tilde{x} - \tilde{x}')}$$

4. (5 pts) Give the equivalent Mahalanobis distance on the original data $\mathbf{x}, \mathbf{x}'$ by specifying A. (Hint: you may need $\hat{\mu}_j, \hat{\sigma}_j$)

   Beginning from the Euclidian distance on the preprocessed data, we see that it can be written as:

$$dist_{Euclidian}(\tilde{x}, \tilde{x}') = \sqrt{(\tilde{x} - \tilde{x}')^\top I(\tilde{x} - \tilde{x}')}$$

   Now,

$$\tilde{x} - \tilde{x}' = [(\frac{x_1 - \hat{\mu}_1}{\hat{\sigma}_1} - \frac{x'_1 - \hat{\mu}_1}{\hat{\sigma}_1}), (\frac{x_2 - \hat{\mu}_2}{\hat{\sigma}_2} - \frac{x'_2 - \hat{\mu}_2}{\hat{\sigma}_2}), ..., (\frac{x_d - \hat{\mu}_d}{\hat{\sigma}_d} - \frac{x'_d - \hat{\mu}_d}{\hat{\sigma}_d})] = [\frac{(x_1 - x'_1)}{\hat{\sigma}_1}, ..., \frac{(x_d - x'_d)}{\hat{\sigma}_d}] \in \mathbb{R}^d$$

   Thus,

$$\sqrt{(\tilde{x} - \tilde{x}')^\top I(\tilde{x} - \tilde{x}')} = \sqrt{(\mathbf{x} - \mathbf{x}')^\top \begin{pmatrix} \frac{1}{\hat{\sigma}_1^2} & 0 & ... & 0 \\ 0 & \frac{1}{\hat{\sigma}_2^2} & ... & 0 \\ 0 & 0 & ... & \frac{1}{\hat{\sigma}_d^2} \end{pmatrix} (\mathbf{x} - \mathbf{x}')}$$

   Thus, if we specify $A = \begin{pmatrix} \frac{1}{\hat{\sigma}_1^2} & 0 & ... & 0 \\ 0 & \frac{1}{\hat{\sigma}_2^2} & ... & 0 \\ 0 & 0 & ... & \frac{1}{\hat{\sigma}_d^2} \end{pmatrix}$, We can say that the following holds:

$$\sqrt{(\tilde{x} - \tilde{x}')^\top I(\tilde{x} - \tilde{x}')} = \sqrt{(\mathbf{x} - \mathbf{x}')^\top A(\mathbf{x} - \mathbf{x}')} = d_A(\mathbf{x}, \mathbf{x}')$$

5. (5 pts) Let the diagonal elements of A be $a_{11}, \ldots, a_{dd}$. Define a diagonal matrix L with diagonal $\sqrt{a_{11}}, \ldots, \sqrt{a_{dd}}$. Define $\tilde{\mathbf{x}} = L\mathbf{x}$. Prove that $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = d_A(\mathbf{x}, \mathbf{x}')$ where I is the identity matrix.

   We notice that $L^\top L = A$. So,

$$d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top I(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')} = \sqrt{(L\mathbf{x} - L\mathbf{x}')^\top I(L\mathbf{x} - L\mathbf{x}')} = \sqrt{(\mathbf{x} - \mathbf{x}')^\top L^\top IL(\mathbf{x} - \mathbf{x}')}$$

   As, $L^\top IL = L^\top L = A$. Thus,

$$d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\mathbf{x} - \mathbf{x}')^\top L^\top IL(\mathbf{x} - \mathbf{x}')} = \sqrt{(\mathbf{x} - \mathbf{x}')^\top A(\mathbf{x} - \mathbf{x}')} = d_A(\mathbf{x}, \mathbf{x}')$$

6. (5 pts) Geometrically, what does $L\mathbf{x}$ do to the point $\mathbf{x}$? Explain in simple English.

   Essentially, since L is a diagonal matrix, it applies a scaling on the i-th component of the data proportional to $\sqrt{a_{ii}}$. So each component gets scaled accordingly, and it produces a linear transformation geometrically, where each of the d-dimensions. This is essentially equal to a Hadamard product of the vectors $\{a_{11}, ..., a_{dd}\}$ and $\{x_1, ..., x_d\}$. Each of the x_i's get scaled and thus, transformed linearly. If all a_{ii}'s were equal, the transformation would be a linear scalar transformation, and it would be as if the magnitude of the vector x was scaled by a in the d-dimensional vector space.

7. (10 pts) Let U be any orthogonal matrix. Define $\tilde{\mathbf{x}} = UL\mathbf{x}$. (i) Prove that $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = d_A(\mathbf{x}, \mathbf{x}')$ again. (ii) Geometrically, what does $UL\mathbf{x}$ do to the point $\mathbf{x}$? Explain in simple English.

   (i) We note that for an orthogonal matrix Q, the following holds: $QQ^\top = Q^\top Q = I$. Thus,

$$d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top I(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')} = \sqrt{(UL\mathbf{x} - UL\mathbf{x}')^\top I(UL\mathbf{x} - UL\mathbf{x}')}$$

$$= \sqrt{(\mathbf{x} - \mathbf{x}')^\top L^\top U^\top IUL(\mathbf{x} - \mathbf{x}')}$$

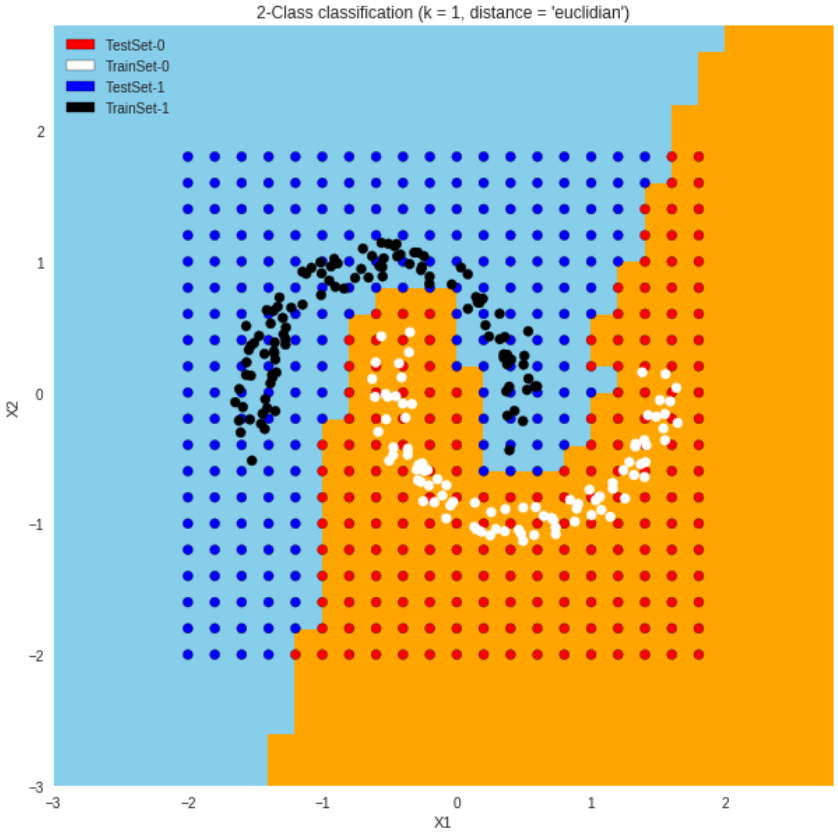
   Now, $L^\top U^\top IUL = L^\top IL = A$, as $U^\top IU = I$ and $L^\top IL = A$ Thus,

$$d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\mathbf{x} - \mathbf{x}')^\top L^\top U^\top IUL(\mathbf{x} - \mathbf{x}')} = \sqrt{(\mathbf{x} - \mathbf{x}')^\top A(\mathbf{x} - \mathbf{x}')} = d_A(\mathbf{x}, \mathbf{x}')$$

   (ii) As discussed before, Lx leads to a Hadamard like product between the square-rooted diagonal elements $\{\sqrt{a_{11}}, ..., \sqrt{a_{dd}}\}$ and $\mathbf{x} \in \mathbb{R}^d$. Proceeding on with this, ULx leads to a linear transformation of Lx that preserves the dot product of Lx and other isometries, i.e. preserves the vector's lengths.

8. (20 pts) Use the whole D2z.txt as training set. Use Euclidean distance (i.e. $A = I$). Visualize the predictions of 1NN on a 2D grid $[-2 : 0.1 : 2]^2$. That is, you should produce test points whose first feature goes over $-2, -1.9, -1.8, \ldots, 1.9, 2$, so does the second feature independent of the first feature. You should overlay the training set in the plot, just make sure we can tell which points are training, which are grid.

In the solution figure, the training set D2z.txt can be seen in the colors black and white. Additionally, the grid extending from -2 to 2 in steps of 0.2 has been assigned a label(colored accordingly) based on the 1NN classification from the training data. Moreover, the whole 9x9 grid from -3 to 3 has been colored to further visualize the decision boundary.



9. (To normalize, or not to normalize?) Start from D2a.txt. Perform 5-fold cross validation.

   (a) (5 pts) Do not normalize the data. Report 1NN cross validation error rate for each fold, then the average (that's 6 numbers).

   (b) (5 pts) Normalize the data. Report 1NN cross validation error rate (again 6 numbers). (Hints: Do not normalize the labels! The relevant quantities should be estimated from the training portion, but applied to both training and validation portions. This should happen 5 times. Also, you would either change $\mathbf{x}$ into $\tilde{x} = Lx$ but then use Euclidean distance on $\tilde{x}$, or do not change x but use an appropriate A; don't mix the two.)

   (c) (5 pts) Look at D2a.txt, explain the effect of normalization on CV error. Hint: the first 4 features are different than the next 2 features.

(a) The 5-fold Cross Validation Error Rates are: [0., 0., 0., 0., 0.]
Average error rate: 0.0

(b) The 5-fold Cross Validation Error Rates for Normalized data: [0.07317073, 0.04878049, 0.125, 0.12820513, 0.1025641]
Average Error Rate: 0.09554409005628517

(c) Picking up on the hint, if we look at the data, we notice that the first four features(or attributes) are essentially irrelevant. We can prove this by running Cross-Validated 1NN predictions on three kinds of dataset: X_{good} which has just the last two attributes. X_{bad} which has the first four attributes. X which is the original data.The average prediction error rates for training on the three different datasets pre and post-normalization are as follows:

Training Dataset	CV-Error	CV Error-Normalized
X	0.0	0.1
X_{good} - only $x5$, $x6$	0.0	0.07
$X_{bad} - x1, x2, x3, x4$	0.55	0.54

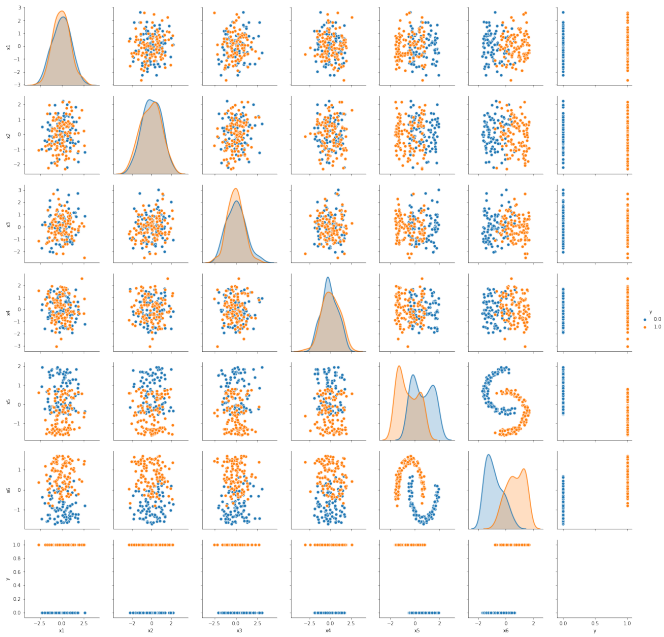
Firstly, this is empirical proof that in the un-normalized dataset, the first four features are irrelevant and have no role at all in predicting labels of the test points. Secondly, since 1NN is essentially a distance-metric based classification method, the classification done using the distance based metric is heavily dominated by the last two features who have the maximum variance, and thus, impact the distance the most as the last two dimensions are analogous to being the Principal Components of the dataset. This significant difference in variance can be seen in the following table that is computed on the un-normalized data.

Attributes	x1	x2	x3	x4	x5	x6
Variance	0.000109	0.000113	0.000091	0.000095	0.863844	0.450556

However, after normalization is carried out, two things happen:

- Due to rescaling of variance across all attributes to 1, the ompact of the last two features on the distance function is slightly diminished and noise is introduced which leads to an increase in error.
- Due to change in variance across attributes, the covariance across the attributes also changes and thus, it might impact distance relations.

This fact can again be visually seen on a pairplot of the normalized data. The pairplot is able to retain geometrical properties of the original data but can give us an idea about the variation in range for all attributes. The range in the two relevant attributes is much greater than the other attributes in the unnormalized dataset, and so these relevant attributes play a greater role in the distance metric and is able to separate points better when compared to the normalized version where the range gets squished.

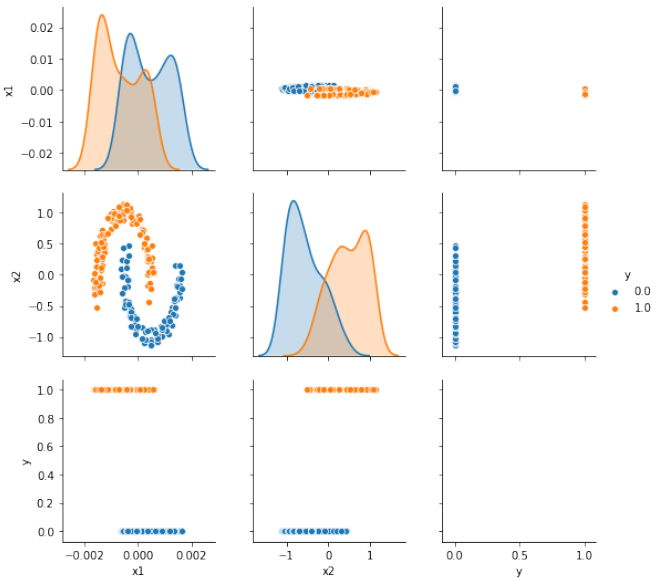


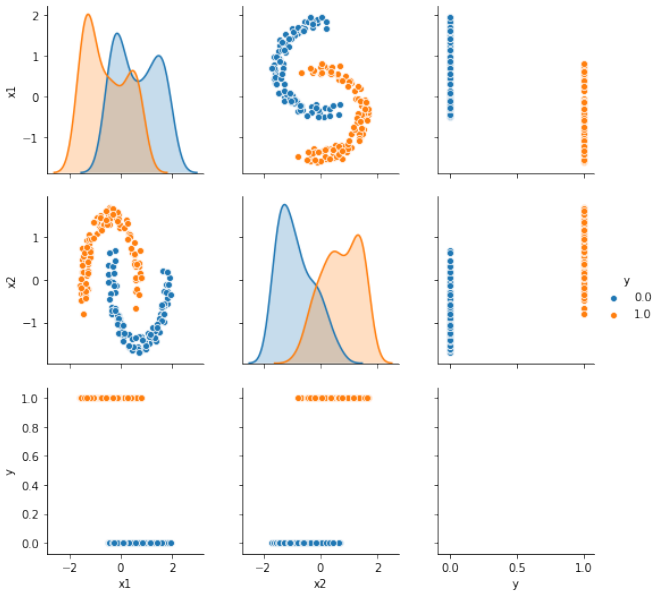
10. (Again. 10 pts) Repeat the above question, starting from D2b.txt.

    (a) Unnormalized: The 5-fold Cross Validation Error Rates are: [0.15, 0.2, 0.15, 0.275, 0.2]
    Average error rate: 0.195

    (b) Normalized: The 5-fold Cross Validation Error Rates are: [0.225, 0.1, 0.125, 0.15, 0.225]
    Average error rate: 0.16

    (c) We can again look at the pairplots of the dataset pre-(plot 1) and post-(plot 2) normalization to see what's happening.





    Looking at the pre-normalization plot, we can clearly see that feature x1 is quite irrelevant in its original form as its spread is not as big as x2, whereas in x2, the classes are distinctly separated and is the principal contributing factor in making the distance-based NN decisions. However, upon normalization, the constriction is removed and the attribute contributes meaningfully well to the classification decision being made that causes a small improvement in the CV test error.

11. (5 pts) What do you learn from Q9 and Q10?

In distance based classification models like the 1NN, the decision is influenced by the measurement units also. When applying 1NN on multi-feature data, where the features have different ranges in their values, most of the distance based decisions will be made based on the features that have a larger range as this as across different samples, these are the dimensions along which there is the possibility of most variation. In the situation where most constricted(low-range) features are irrelevant(like in D2a), it doesn't make sense to normalize the data as that leads to dampening of the influence that the relevant features had on the distance based decision making. However, in cases where all attributes are more or less relevant(D2b), normalization can help improve the performance by rescaling one or more of the features. So, essentially, when the nearest neighbor decision is made on the basis of few the Principal components, it can really hurt to normalize if there are other irrelevant features in the dataset.

12. (Weka, 10 pts) Repeat Q9 and Q10 with Weka. Convert appropriate data files into ARFF format. Choose classifiers / lazy / IBk. Set $K = 1$. Choose 5-fold cross validation. Let us know what else you needed to set. Compare Weka's results to your Q9 and Q10.

    (I) D2a
    (i) The CV-error for unnormalized data: 0.05
    (ii) The CV error for normalized data: 0.05

    (II) D2b
    (i) The CV-error for unnormalized data: 0
    (ii) The CV error for normalized data: 0