HOMEWORK 3

$AKASHSHARMA$
9081731771

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?
   For a diagonal matrix A to be a positive semidefinite diagonal matrix, the eigen values of A must be greater than or equal to 0. Since, the eigen values of the diagonal matrix conincide with the diagonal entries, the diagonal elements of the diagonal matrix must be greater than or equal to 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.)
   We can preprocess the data by subtracting the mean of each feature (all feature values from the dataset) from the feature value, and dividing by the respective variance of each feature as well.

3. (5 pts) Let $\tilde{\mathbf{x}}$ be the preprocessed data. Give the formula for the Euclidean distance between $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}'$.
Since, $\tilde{\mathbf{x}}$ is the preprocessed data, each of the values of the original dataset is normalized, that is, the mean is subtracted, and values are divided by the standard deviation of each feature.

$$Euclidean(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{\sum_i (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}'_i)^2}$$

Here, each value of feature $\tilde{\mathbf{x}}_i$ is,

$$(x_i - \mu_i)/\sigma_i$$

where $\mathbf{x}_i$ is the original feature, μ_i is the mean, and σ_i is the standard deviation of the i_{th} feature.

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

$$d_A(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_A = \sqrt{(\mathbf{x} - x')^\top A(\mathbf{x} - x')}.$$

where

$$A = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{dd} \end{bmatrix}$$

Putting the value of A in the above equation and mutiplying the matrices, we get

$$A = \sqrt{\sum_i a_{ii}(x_i - x'_i)^2}$$

Now, comparing this with the above Euclidean distance,

$$Euclidean(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{\sum_i (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}'_i)^2}$$

Here, each value of feature $\tilde{\mathbf{x}}_i$ is,

$$(x_i - \mu_i)/\sigma_i$$

where $\mathbf{x}_i$ is the original feature, μ_i is the mean, and σ_i is the standard deviation of the i_{th} feature.
Replacing the value of $\tilde{\mathbf{x}}_i$, that is,

$$(x_i - \mu_i)/\sigma_i$$

above we get,

$$\sqrt{\sum_i 1/(\sigma_i)^2(\mathbf{x}_i - \mathbf{x}'_i)^2}$$

Therefore, on comparing the equations above, the PSD matrix must have each of its Diagonal values as $1/\sigma_i^2$. The equivalent Mahalonobis distance is

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

, where

$$A = \begin{bmatrix} 1/(\sigma_1)^2 & 0 & 0 & \dots & 0 \\ 0 & 1/(\sigma_2)^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1/(\sigma_d)^2 \end{bmatrix}$$

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.

   As L is a diagonal matrix with diagonal $\sqrt{a_{11}}, \ldots, \sqrt{a_{dd}}$,

$$L = \begin{bmatrix} \sqrt{a_{11}} & 0 & 0 & \ldots & 0 \\ 0 & \sqrt{a_{22}} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \sqrt{a_{dd}} \end{bmatrix}$$

   As, $\tilde{\mathbf{x}} = L\mathbf{x}$, so

$$\tilde{\mathbf{x}} = \begin{bmatrix} \sqrt{a_{11}}x_1 \\ \sqrt{a_{22}}x_2 \\ \vdots \\ \sqrt{a_{dd}}x_d \end{bmatrix}$$

   $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top A(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')}$
   Since, A = Identity, $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top (\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')} = \sqrt{\sum_i a_{ii}(x_i - x_i')^2}$

   Looking at the RHS of the equation,

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

$$A = \begin{bmatrix} a_{11} & 0 & 0 & \ldots & 0 \\ 0 & a_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & a_{dd} \end{bmatrix}$$

   the values of L are the square root of those in A. On expanding the above, we get LHS = RHS = $\sqrt{\sum_i a_{ii}(x_i - x_i')^2}$. Hence LHS = RHS.

6. (5 pts) Geometrically, what does $L\mathbf{x}$ do to the point $\mathbf{x}$? Explain in simple English.
   The Matrix L when multiplied with x, results in increasing or decreasing the values of each feature of x, resulting it to move in the feature space.

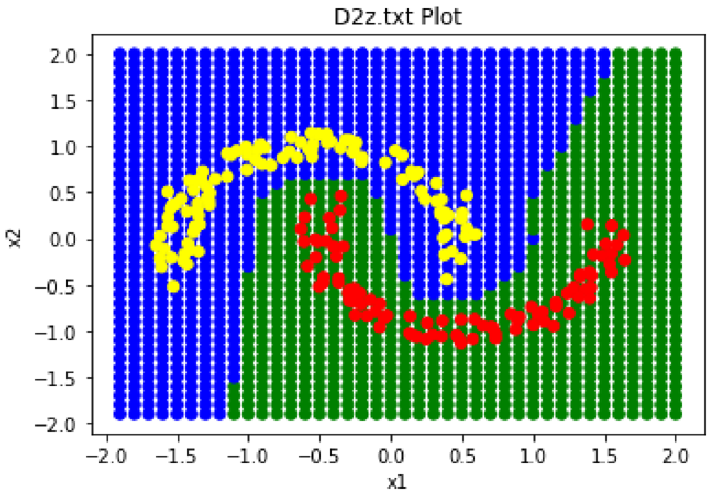
7. (10 pts) Let U be any orthogonal matrix. Define $\tilde{\mathbf{x}} = UL\mathbf{x}$.
   (i) Prove that $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = d_A(\mathbf{x}, \mathbf{x}')$ again.
   Since, U is an Orthogonal Matrix, $U_T U = I$. Expanding the LHS with A = I, and $\tilde{\mathbf{x}} = UL\mathbf{x}$
   $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top (\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')} = \sqrt{(UL\mathbf{x} - UL\mathbf{x}')^\top (UL\mathbf{x} - UL\mathbf{x}')}$
   $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(UL)^T (\mathbf{x} - \mathbf{x}')^\top UL(\mathbf{x} - \mathbf{x}')}$
   Using the property, $(UL)^T = L^T U^T$, and using the Orthogonal property, $U^T U = I$, and $L^T L = L^2$ in the equation below we get,
   $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\mathbf{x} - \mathbf{x}')^\top L^2 (\mathbf{x} - \mathbf{x}')}$ Since, $L^2 = A$, we get
   $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = d_A(\mathbf{x}, \mathbf{x}')$

   (ii) Geometrically, what does $UL\mathbf{x}$ do to the point $\mathbf{x}$? Explain in simple English.
   $UL\mathbf{x}$ when multiplied to a point $\mathbf{x}$ results in the linear transformation of the vector, such as rotation, reflection or rotoreflection. Its a unitary transformation.

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.

   The plot below shows the grid (test set) and the training set along with the labels. Here, red denotes the points with label 0 in the training set, while yellow denotes the points with label 1 in the training set (D2z.txt). While, green denotes the points with label 0 in the testing set, and blue denotes the points with label 1 in the testing set.

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



D2z Scatter Plot

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

For, D2a.txt, without normalizing the data, the 1NN cross validation error rate is 0. The values for each fold is: [0.0, 0.0, 0.0, 0.0, 0.0]. The average 1NN cross validation error rate is also 0.

(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 $\tilde{\mathbf{x}} = L\mathbf{x}$ but then use Euclidean distance on $\tilde{\mathbf{x}}$, or do not change $\mathbf{x}$ but use an appropriate A; don't mix the two.)

For, D2a.txt, after normalizing the data, the values for each fold is: [0.13, 0.06, 0.1, 0.17, 0.08]. The average 1NN cross validation error rate is 0.108.

(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.

After Normalizing the dataset, the cross validation error is increasing for D2a.txt. In this dataset, all features are not contributing equally to the classification of labels. The scales of the features are also different. So, when we normalise the data, the scales become same for all the features and now are contributing equally.

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

(a) For, D2b.txt, without normalizing the data, the values for each fold is: [0.16, 0.18, 0.15, 0.28, 0.17]. The average 1NN cross validation error rate is 0.188.

(b) For, D2b.txt, after normalizing the data, the values for each fold becomes 0, that is: [0.0, 0.0, 0.0, 0.0, 0.0]. The average 1NN cross validation error rate is also 0.0.

(c) After Normalizing the dataset, the cross validation error is decreasing for D2b.txt. In this dataset, before normalisation all features are not contributing equally to the classification of labels. The scales of the lables are also different. So, when we normalise the data, the scales become same for all values and now are contributing equally, which is favourable in this case, as the error is decreasing.

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

So, in Q9 and Q10, we see different effects of normalisation on the cross validation error. In Q9, after normalisation, the error is increasing, and in Q10, it is decreasing. So, we can say that normalisation always won't lead to a decrease in cross validation errors. In certain datasets, the output might depend more on one of the features, while it might depend less on other features. So, if we normalise the dataset, it will scale all features equally, which might lead to an increase or decrease in cross validation errors.

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.

For this question, we also had to set the values of normalise parameter to True or False, and had to keep the K values as 1, and fold value as 5.

For D2a.txt, the value for cross validation error without normalisation is 0. After normalisation, the value of cross validation becomes 0.06. So, this value is lower as compared to my cross validation error computed in Q9.

For D2b.txt, the value for cross validation error after normalisation is 0. Before normalisation, the value of cross validation error is 0.10.