HOMEWORK 3

>>Abhay Kumar<<
>>9081403157<<

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:

$$x_{11} \quad \dots \quad x_{1d} \quad y_1$$
$$\dots$$
$$x_{n1} \quad \dots \quad x_{nd} \quad y_n$$

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?
let the diagonal matrix, $A = \begin{bmatrix} A_{11} & 0 & 0 & \dots & 0 \\ 0 & A_{22} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & A_{dd} \end{bmatrix}$
For A to be PSD, there exists a v such that $v^T Av \geq 0$ for all v in $\mathbb{R}^d$.
$\implies \sum_{j=1}^{d} A_{jj} v_j^2 \geq 0$
$\implies A_{jj} \geq 0 \ \ \forall\, j$ (can be proved by contradiction if we set v to be such that $v_j \neq 0$ and $v_k = 0 \ \ \forall\, k \neq j$, then we need to have diagonal elements, $A_{jj} \geq 0$ for A to be PSD.

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.)
Notations: $x_{ij} = j^{th}$ dimension of i^{th} example from training data set D
for each dimension $j = 1, 2, \ldots, d$
$\hat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$

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

preprocess: $\tilde{\mathbf{x}}_{ij} \leftarrow \frac{x_{ij} - \hat{\mu}_j}{\hat{\sigma}_j}$

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

Euclidean distance, $d(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\tilde{\mathbf{x}} - \tilde{x}')^\top(\tilde{\mathbf{x}} - \tilde{x}')}$, where $\tilde{\mathbf{x}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ...\tilde{\mathbf{x}}_j, ...\tilde{\mathbf{x}}_d]^T$ and $\tilde{\mathbf{x}}' = [\tilde{\mathbf{x}}'_1, \tilde{\mathbf{x}}'_2, ...\tilde{\mathbf{x}}'_j, ...\tilde{\mathbf{x}}'_d]^T$
$= \sqrt{\sum_{j=1}^{d}(\tilde{\mathbf{x}}_j - \tilde{x}'_j)^2}$
$= \sqrt{\sum_{j=1}^{d}(\frac{\mathbf{x}_j - \hat{\mu}_j}{\hat{\sigma}_j} - \frac{x'_j - \hat{\mu}_j}{\hat{\sigma}_j})^2}$ (using the pre-processing normalization as mentioned in above question)
$= \sqrt{\sum_{j=1}^{d}(\frac{\mathbf{x}_j - x'_j}{\hat{\sigma}_j})^2}$

4. (5 pts) Give the equivalent Mahalanobis distance on the original data $\mathbf{x}, \mathbf{x}'$ by specifying A. (Hint: you may need $\hat{\mu}_j, \hat{\sigma}_j$)
Euclidean distance, $d(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')$ can be written equivalently as $d_A(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - x')^\top A(\mathbf{x} - x')}$,
where the diagonal matrix, $A = \begin{bmatrix} \frac{1}{\hat{\sigma}_1^2} & 0 & 0 & \ldots & 0 \\ 0 & \frac{1}{\hat{\sigma}_2^2} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \frac{1}{\hat{\sigma}_d^2} \end{bmatrix}$ (comparing the above expanded version in Q3)

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.

Let the diagonal matrix, $A = \begin{bmatrix} a_{11} & 0 & 0 & \ldots & 0 \\ 0 & a_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & a_{dd} \end{bmatrix}$
then the diagonal matrix, $L = \begin{bmatrix} \sqrt{a_{11}} & 0 & 0 & \ldots & 0 \\ 0 & \sqrt{a_{22}} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \sqrt{a_{dd}} \end{bmatrix}$
Now, $d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top I(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')}$ where I is the identity matrix.
$d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(L\mathbf{x} - L\mathbf{x}')^\top I(L\mathbf{x} - L\mathbf{x}')} = \sqrt{\sum_{j=1}^{d}(\sqrt{a_{jj}}(\mathbf{x}_j - x'_j))^2}$
$= \sqrt{\sum_{j=1}^{d} a_{jj}(\mathbf{x}_j - x'_j)^2}$
$= \sqrt{(\mathbf{x} - x')^\top A(\mathbf{x} - x')} = \|\mathbf{x} - \mathbf{x}'\|_A$
$= d_A(\mathbf{x}, \mathbf{x}')$

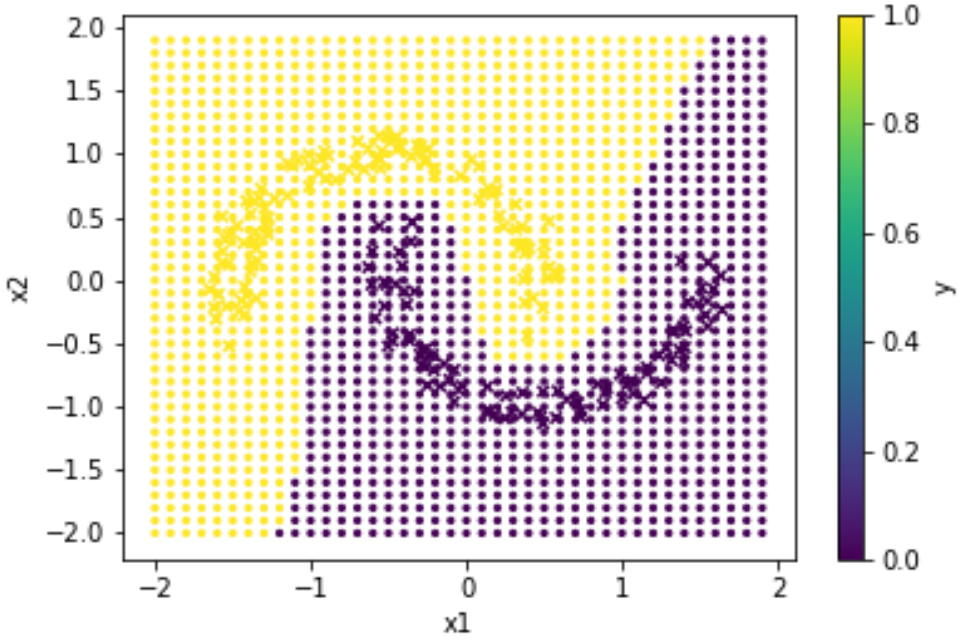
6. (5 pts) Geometrically, what does $L\mathbf{x}$ do to the point $\mathbf{x}$? Explain in simple English.
$L\mathbf{x}$ scales the j^{th} dimension of $\mathbf{x}$ by $\sqrt{a_{jj}}$. Since $\sqrt{a_{jj}} \geq 0$, the direction of vector $\mathbf{x}$ will be retained. when $\sqrt{a_{jj}} > 1$ the j^{th} dimension of $\mathbf{x}$ is dilated, and if $\sqrt{a_{jj}} < 1$ then the j^{th} dimension of $\mathbf{x}$ is contracted.

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.
$d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top I(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')}$ where I is the identity matrix.
$d_I(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sqrt{(UL\mathbf{x} - UL\mathbf{x}')^\top I(UL\mathbf{x} - UL\mathbf{x}')}$

$= \sqrt{(\mathbf{x}-\mathbf{x}')^\top L^\top U^\top U L(\mathbf{x}-\mathbf{x}')} \quad (putting\ U^\top U = I)$
$= \sqrt{(\mathbf{x}-\mathbf{x}')^\top L^\top L(\mathbf{x}-\mathbf{x}')} \quad (putting\ L^\top L = A)$
$= \sqrt{(\mathbf{x}-x')^\top A(\mathbf{x}-x')} = \|\mathbf{x}-\mathbf{x}'\|_A$
$= d_A(\mathbf{x},\mathbf{x}')$

Geometrically, $U\mathbf{x}$ reflects $\mathbf{x}$ in some plane i.e. rotates it. $UL\mathbf{x}$ will scale and rotate the features

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.



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).
   error rate $= \frac{Number\ of\ failed\ classification}{total\ number\ of\ test\ examples}$

	error rate
fold 1	0
fold 2	0
fold 3	0
fold 4	0
fold 5	0
average	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.)

	error rate
fold 1	0.125
fold 2	05
fold 3	0.1
fold 4	0.15
fold 5	0.175
average	0.1

(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.

The first 4 features are very small numbers close to zero and randomly distributed across different class labels. Whereas, the last 2 features are the dominants and highly separable for different class labels.

Normalizing the first 4 features increases the CV error, because of increased contribution of these features in the distance calculation. Normalization causes dilation of all these 4 features and therefore adding more noise to distance metric.

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

Before Normalization

	error rate
fold 1	0.175
fold 2	0.15
fold 3	0.225
fold 4	0.225
fold 5	0.175
average	0.19

After normalization

	error rate
fold 1	0
fold 2	0
fold 3	0
fold 4	0
fold 5	0
average	0

Effect of normalization:
One axis has all points lying in between -0.002 and 0.002. even though all points are separable along this axis, the distance formulation in this space almost ignored the distance along this axis. Normalizing both axis makes both features to contribute effectively in the distance formulation and hence, both axis were equally accounted for the classification purpose.

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

Normalization should be thought out carefully depending on the importance of that feature in classification. We should ask to domain experts about the physical significance of all features and the features which just add to the noise in dataset for the given classification task.

Be aware of the different features available. Some features could be just noise and not relevent for the classification and normalizing these features blindly may end up giving worse performance (i.e. less accuracy).

If different axis/features seems to be equally important for the classification, normalizing all these axis should increase the classification performance.

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.

5-fold CV error rate from WEKA:-

	weka error rate	my error rate
D2a	0	0
D2a (Normalized)	0.08	0.1
D2b	0.195	0.19
D2b (Normalized)	0	0

I need to set the following-
weka.classifiers.lazy.IBK - nearestNeighbourSearchAlgorithm - LinearNNSearch - distanceFunction - EuclideanDistance - dontNormalize = True

Compare Weka's results to your Q9 and Q10.
The result trends are similar for both Q9 and Q10 before and after normalization for both datasets (D2a and D2b). The average error rate is almost same or similar, difference maybe because of different 5-folds splits done in my code and weka implementation