100062236林修安 100062201王俊智

Machine Leaning Final Project Report

Algorithm Design

1. Feature Selection
   - Plot 2D figures by all combinations of every two attributes as x-axis and y-axis respectively.

(1) Fig.1 is the plot with respect to the first and second attribute, and it's obvious that the relationship between the two attributes can be dealt as two-moon data.

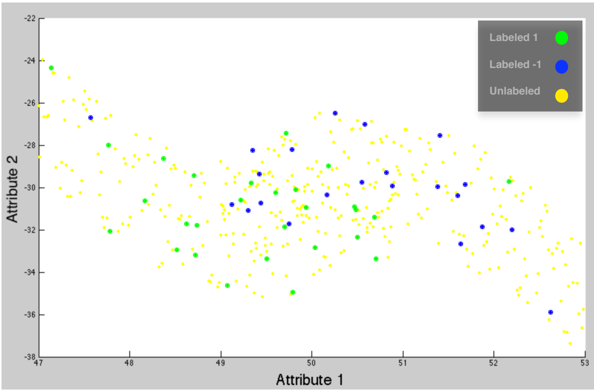

Fig.1

(2) Fig.2 is the plot with respect to the first and third attributes, and it shows a slight tendency that the relationship between them can be interpreted by a linear model.

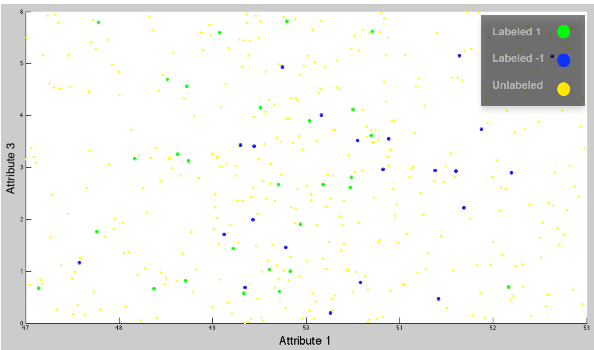

Fig.2

(3) Fig.3 is the plot with respect to the fourth and fifth attributes, and there is no obvious relationship between the two attributes.

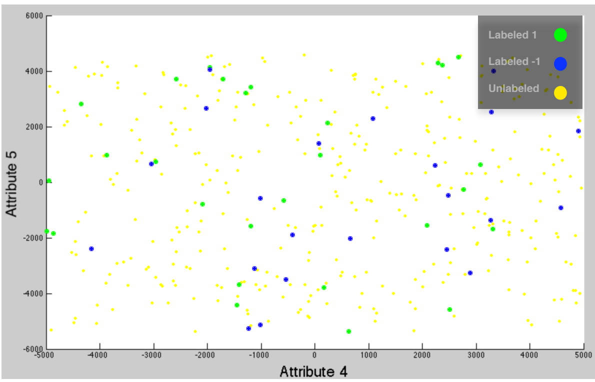

Fig.3

- Use different classification models to train all combinations of every two and three attributes, and record the testing accuracy. Accuracy reaches the first baseline is shown in red.

(1) SoftMarginLinearClassifier

SoftMarginLinearClassifier	
Training Dimension	Accuracy
1 2	0.9
1 3	0.8
1 4	0.8
1 5	0.78
2 3	0.6
2 4	0.48
2 5	0.46
3 4	0.58
3 5	0.56
4 5	0.54

Cht.1

SoftMarginLinearClassifier	
Training Dimension	Accuracy
1 2 3	0.82
1 2 4	0.82
1 2 5	0.84
1 3 4	0.7
1 3 5	0.82
1 4 5	0.78
2 3 4	0.64
2 3 5	0.58
2 4 5	0.48
3 4 5	0.52

Cht.2

The result of SoftMarginLinearClassifier with combinations of any two attributes shows that the first attribute have high dependency with all the other attributes, while the other combinations fail to interpret the relationship.

The result of SoftMarginLinearClassifier with combinations of any three attributes shows that the relationship of such data depends highly on the first attribute.

(2) LapSVMClassifier

LapSVMClassifier	
Training Dimension	Accuracy
1 2	0.88
1 3	0.86
1 4	0.8
1 5	0.8
2 3	0.64
2 4	0.5
2 5	0.48
3 4	0.56
3 5	0.5
4 5	0.54

Cht.3

LapSVMClassifier	
Training Dimension	Accuracy
1 2 3	0.94
1 2 4	0.78
1 2 5	0.82
1 3 4	0.72
1 3 5	0.8
1 4 5	0.72
2 3 4	0.52
2 3 5	0.66
2 4 5	0.48
3 4 5	0.46

Cht.4

LapSVMClassifier reveals another dependency that there’s a certain relation between the second and third attributes.

(3) LapRLSClassifier

LapRLSClassifier	
Training Dimension	Accuracy
1 2	0.82
1 3	0.7
1 4	0.24
1 5	0.16
2 3	0.66
2 4	0.24
2 5	0.14
3 4	0.24
3 5	0.16
4 5	0

Cht.5

LapRLSClassifier	
Training Dimension	Accuracy
1 2 3	0.9
1 2 4	0.26
1 2 5	0.22
1 3 4	0.26
1 3 5	0.22
1 4 5	0
2 3 4	0.26
2 3 5	0.22
2 4 5	0
3 4 5	0

Cht.6

LapRLSClassifier narrows the dimensionality down to attributes 1, 2, and 3. As shown in Cht.5, it's obvious that attributes 1, 2, and 3 are highly related. In Cht.6, such dependency is again confirmed by the outperforming accuracy.

- As our conclusion, we only utilize the first three attributes in training and classification. Fig.4 and Fig.5 below are the result of plotting 3D figures according to the first three attributes from a certain perspective, and we found that it's two-moon data.

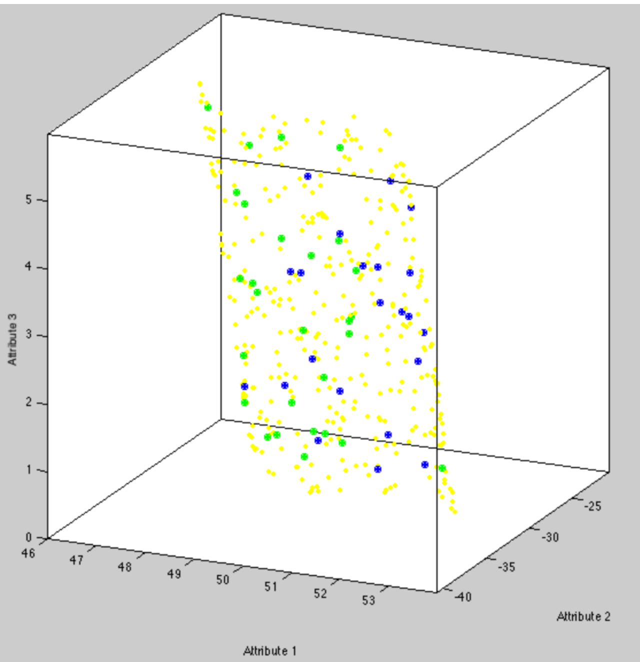

Fig.4

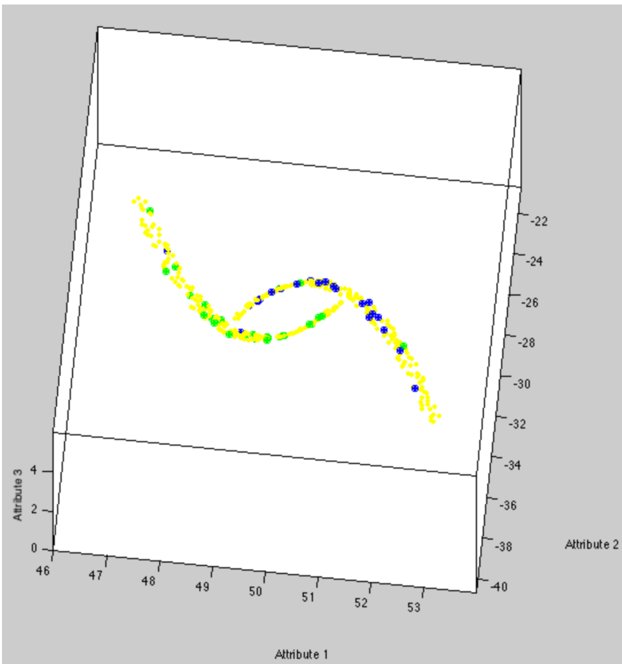

Fig.5

2. Dimensionality Reduction - PCA
   - PCA Implementation Steps
   (1) Find the mean of data
   (2) Subtract data by its mean
   (3) Find the covariance of such data
   (4) Find eigenvalues and eigenvectors of the covariance matrix above
   (5) Arrange the eigenvalues in ascending order with corresponding eigenvectors
   (6) Obtain transMat, the result after PCA, by choosing the first three eigenvectors (reducing to three dimensions)

   - After applying PCA to X and Xtest, we reduce the dimensionality from five to three.

$$X \leftarrow X \times transMat(\mathbb{R}^{400\times5} \rightarrow \mathbb{R}^{400\times3})$$
$$Xtest \leftarrow Xtest \times transMat(\mathbb{R}^{50\times5} \rightarrow \mathbb{R}^{50\times3})$$

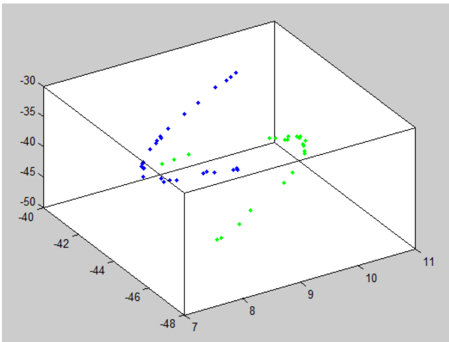

Fig.6
Xtest after PCA

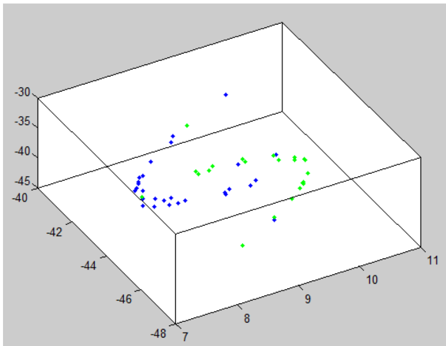

Fig.7
X after PCA

3. Data Preprocessing
   - We found that in the training data, there are some noise. By excluding them, we can achieve better accuracy in classification. The noise are denoted by red circles in Fig.8.

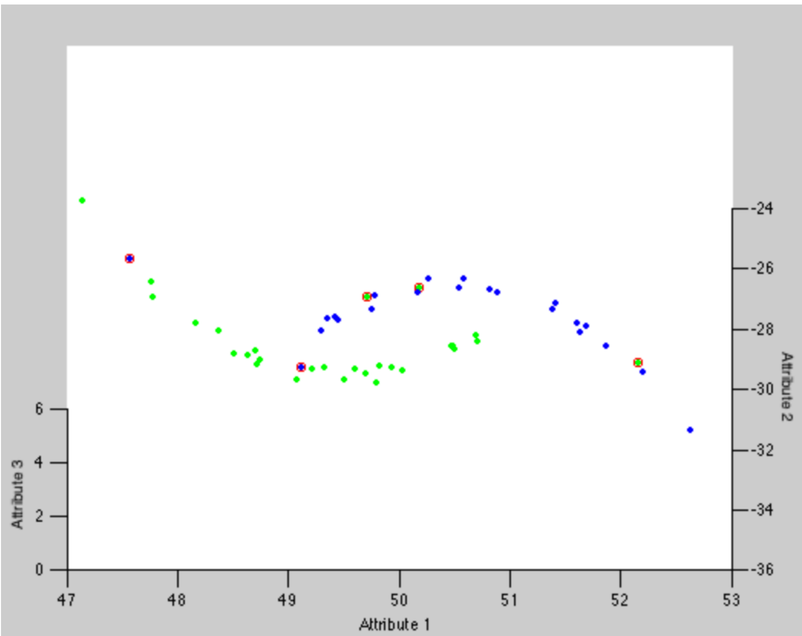

Fig.8

4. Main Algorithm
   - Objective :

$$\arg\min_{\alpha} \| \boldsymbol{r} - JK\alpha \|^2 + \mu\alpha^\top K^\top LK\alpha + \lambda\alpha^\top K\alpha$$

   - By taking partial derivative of α, we obtain the following, and it's used in the LapRLS algorithm for solution.

$$\alpha = (JK + \mu LK + \lambda I)^{-1} J^\top r$$

$\boldsymbol{x} \leftarrow PCA(X)$	▷ Dimensionality Reduction
$\tilde{\boldsymbol{x}} \leftarrow \{\boldsymbol{x}_{not-noise}\}$ $\tilde{\boldsymbol{y}} \leftarrow \{\boldsymbol{y}_{not-noise}\}$	▷ Denoise
$S_{ij} \leftarrow \exp\left(\frac{-\| \tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j \|^2}{\sigma^2}\right)$	▷ Gaussian Similarity Matrix
$D_i \leftarrow \sum_{j=1}^{\tilde{n}} s_{ij}$ $L \leftarrow D - S$	▷ Graph Laplacian Matrix
$J \leftarrow diag(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{\tilde{n}})$ $\alpha \leftarrow (JK + \mu LK + \lambda I_{\tilde{n}})^{-1} J\tilde{y}$	▷ Obtain α

Cross Validation, Training Error & Testing Error

To find the best hyperparameters, we apply leave-one-out cross validation. Our objective is to find the hyperparameters such that minimizes both validation error and testing error.

validation error = 0.022 (1 misclassified)
testing error = 0.04 (2 misclassified)
training error = 0.1 (5 noisy instances misclassified)

Since we eliminate the noise (5 instances), the optimal training error is 0.1.

Before denoising, the testing error is 0.12 and validation error is 0.22. For the rest 350 unlabeled instances, we can plot the result Fig.9 of classification, and it's obvious that many of them are misclassified.

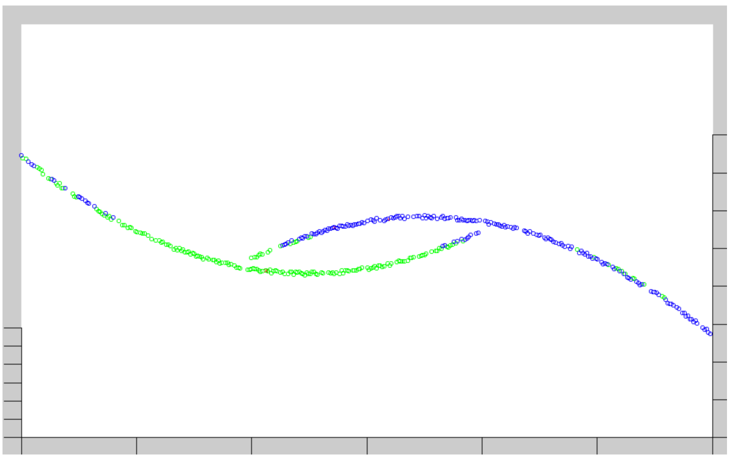

Fig.9

The result of testing is drawn as in Fig.10. Points with red cross are misclassified.

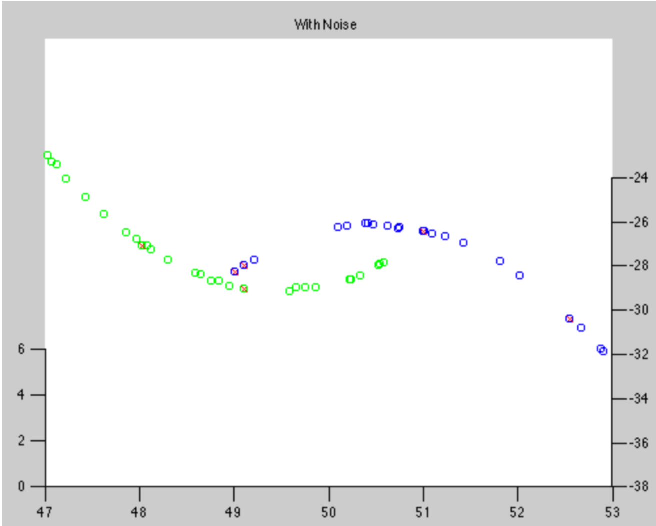

Fig.10

After denoising, the testing error drops to 0.04, and validation error drops to 0.022. The plot Fig.11 of the unlabeled instances is more reasonable.

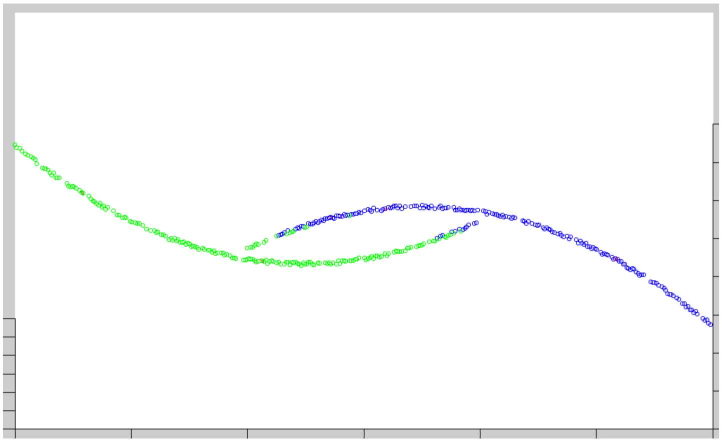

Fig.11

The result of testing is drawn as in Fig.12, and it's also our final result. The red-crossed points are too close to the other class so they are easily misclassified.

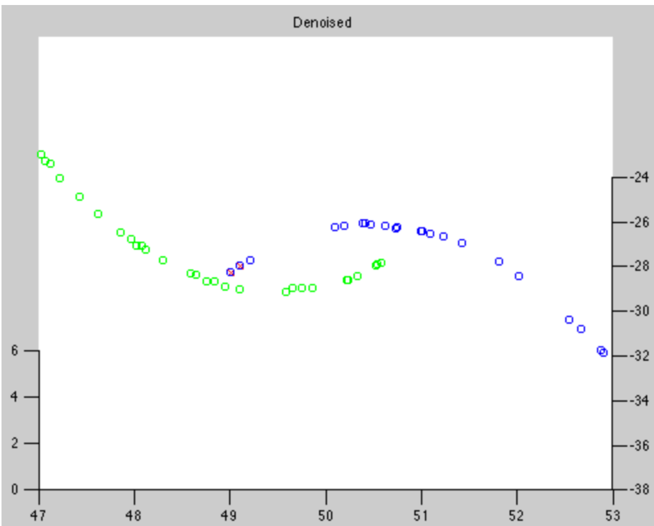

Fig.12