1. Supervise Learning

Instances that are co-owned by supervise algorithms:

1. Matrix X\_orig, Y\_orig, the matrixs are the datasets in the input file
2. Matrix X\_rand, Y\_Rand, if you have randomized your datasets, then the 2 matrixs will be used to keep the randomized X\_orig, Y\_orig, otherwise they will be empty
3. Matrix X\_train, Y\_train, since you might have set the train\_rate to be smaller than 1.0, then X\_train will only keep the front train\_rate of the original datasets or randomized datasets.
4. Bool bRand, be used to judge the input datasets should be randomized.
5. SUPERVISE\_PARA, it is used to set parameters to control the training process.
6. When predict data, at first make the data that you want to predict just as the input, like normalization, feature scaling.

class SUPERVISE\_PARA

{

public:

SUPERVISE\_PARA();

~SUPERVISE\_PARA();

SUPERVISE\_PARA& operator=(SUPERVISE\_PARA& p);

SUPERVISE\_PARA(SUPERVISE\_PARA& p);

public:

double converge; //control when to converge

//prepressoring paras

bool bNorm;

bool bScale;

int dim;

//paras to control train process

int num\_iters;

double alpha;

double lambda;

NNS\_CreateInfo nn\_struct;

//svm

KERNEL\_TYPE kernel\_type;

double tol;

double C; //same as 1/lambda

int train\_method; //to choose which training method to use

};

* 1. Linear Regression

Use X\_train, Y\_train to train data, have no multi-label realization.

Use theta to store the weight. It is initialized with zeros.

Y = theta\*x

Set follows:

Lambda

Alpha,

Converge

Get Follows:

theta

Evaluation: only with Jtrain, Jcv, Jtest

* 1. Logistic Regression

Use X\_mul[index], Y\_train to train data, since it might be used to train multi-label datasets.

Use theta\_mul[index] to store the weight. They are all initialized with zeros.

Evaluation: with Jtrain, Jcv, Jtest, True Positives, False Positives, True Negatives, False Negatives, Accuracy, Precision, Recall, PR.

Y = theta\*h(x)

Set follows:

Lambda

Alpha,

Converge

Get Follows:

theta

* 1. Neural Network

Use X\_train, Y\_train to train data, have no multi-label realization. Since it support the multi-label training.

Use theta to store the weight. It is initialized with randomized small float number.

Evaluation: with Jtrain, Jcv, Jtest, True Positives, False Positives, True Negatives, False Negatives, Accuracy, Precision, Recall, PR.

Y = theta\*h(x)

Set follows:

Lambda

Alpha,

Converge

Neural Network structures info.

Get Follows:

theta

* 1. Support Vector Machine(SVM) (actually belong to logistic regression)

Y = wx+b

Set follows:

C，= same as 1/lambda

Tol

Get Follows:

B

W

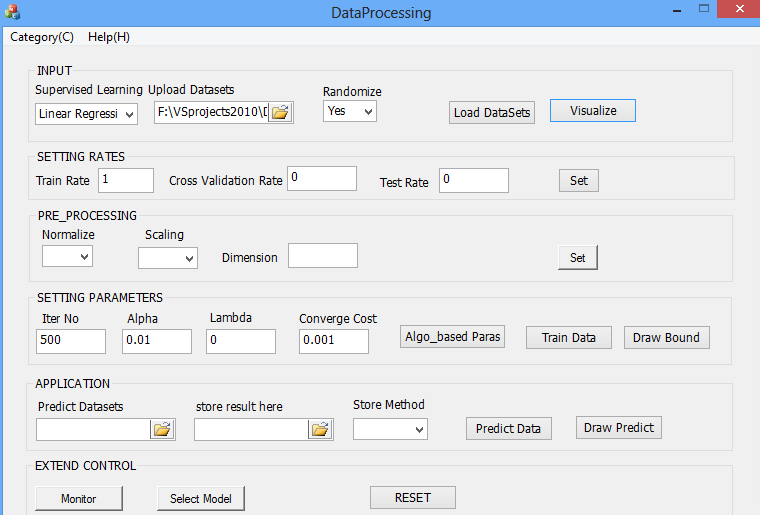
Training will get w and b

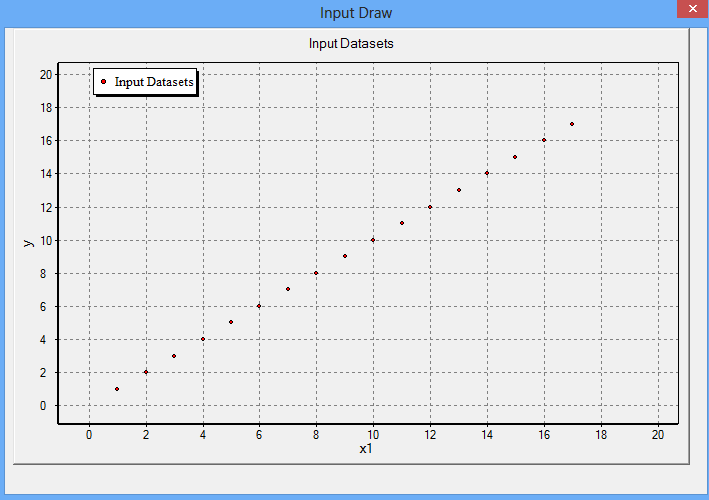
Use X\_mul[index], Y\_train to train data, since it might be used to train multi-label datasets.

Use theta\_mul[index] to store the weight. They are all initialized with zeros.

Evaluation: with Jtrain, Jcv, Jtest, True Positives, False Positives, True Negatives, False Negatives, Accuracy, Precision, Recall, PR.

1. Clustering
2. Examples
   1. Linear regression
3. Choose learning method – Linear Regression, then choose a file used to learn. The example file locate at Data directory, named “lin\_regression.txt”. If you choose “Randomize”, the training examples’ order in the loaded training sets will be randomized. Then put the “Load DataSets” button. After you have successfully loaded the datasets, you could put the “Visualize” button to visualize the training datasets.





1. SETTING RATES

Train rates, cross validation rates, and test rate should sum up to 1.

1. PRE\_PRCESSING

Normalization =

Feature scaling:

X0, x1, x2, to 6 dimentions

H = 1+x1+x2+ x1x2+ x1^2+x2^2

+ x1^3+x1x2^2+x2x1^2+x2^3 ……

1. SETTING PARAMTERS

After you have set the parameters, put the “Train Data” button, then “draw decision boundary”.

1. APPLICATION

If you leave the Predict Datasets blank, then it will use the input file. You can choose a new file, if you chose the MIX method, the result will be attached after the original data every column. If the PURE method, the result will be here.

1. You can check the detail logs in the Directory: Logs/ Supervise
   1. Neural network