Introduction to Data Mining

Lab 4: More Classifiers

## Name: Phạm Đức Đạt

ID: ITITIU20184

## Classification boundaries

In the fifth class, we are going to look at some machine learning methods used to classify datasets in Weka. (See the lecture of class 4 by Ian H. Witten, [1][[1]](#footnote-1)). We are going to learn about linear regression, classification by regression, and support vector machines.

In this section, we are going to start by looking at classification boundaries for different machine learning methods. We are going to use Weka’s **Boundary Visualizer**, and a 2-dimensional datasets. Follow the instructions in [1] to do some experiments, and then fill in the following table with the **classifier models**.

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas**  **et** | **Rules 🡪 OneR** | **Lazy 🡪 IBk** | |
| K=5 | K=20 |
| Iris.2D  .arff | === Classifier model (full training set) ===  petalwidth:  < 0.8 -> Iris-setosa  < 1.75 -> Iris-versicolor  >= 1.75 -> Iris-virginica  (144/150 instances correct) | IB1 instance-based classifier  using 5 nearest neighbour(s) for classification  Time taken to build model: 0 seconds | === Classifier model (full training set) ===  IB1 instance-based classifier  using 20 nearest neighbour(s) for classification |

Try other learning methods, e.g NaiveBayes using SupervisedDiscretization, i.e. supervised discretization is to take the classes into account when discretizing numeric attributes into ranges... [Refer to Text [2]. Chapter 7 for discretization part]

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Bayes > NaiveBayes** | **Trees > J48** | |
| **minNumbObj = 5** | **minNumbObj = 10** |
| Iris.2D.arff |  |  |  |

## Linear regression

In this section, we are going to deal with numeric classes using a classical statistical method.

Follow the lecture of [linear regression](http://en.wikipedia.org/wiki/Linear_regression) in [1] to learn how to calculate weights of attributes from training data, and make predictions. [Refer to Text [2]. Chapter 4.6 for linear regression part]

Follow the instructions in [1] to examine the model of **linear regression** on the **cpu** dataset.

Write down the results in the following table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dataset** | **Correlation coefficient** | **Mean absolute error** | **Root mean squared error** | **Relative absolute error** | **Root relative squared error** |
| Cpu | 0.9012 | 41.0886 | 69.556 | 42.6943 % | 43.2421 % |
| Linear Regression Model |  | | | | |

Do again to examine **M5P** on the **cpu** dataset, and then write down the results in the following table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dataset** | **Correlation coefficient** | **Mean absolute error** | **Root mean squared error** | **Relative absolute error** | **Root relative squared error** |
| Cpu | 0.9274 | 29.8309 | 60.7112 | 30.9967 | 37.7434 |
| Classifier model | M5 pruned model tree: | | | | |
| Linear regression models: | | | | |

Is **M5P** non-linear regression? - – No, M5P is a tree model

## Classification by regression

Follow the instructions in [1] to investigate two‐class classification by regression, using the **diabetes** dataset.

We are going to convert the nominal class to the numeric class so that the linear regression model is applicable.

Write down the results in the following table:

|  |  |
| --- | --- |
| **Classifier model** | **Evaluation** |
|  |  |

## Support vector machines

Learn about logistic regression in [2]. Chapter 4.6

Follow the lecture of support vector machines (SVMs) in [1], …

**Support vector machines** (SVMs, also **support vector networks** [[1]](http://en.wikipedia.org/wiki/Support_vector_machine#cite_note-CorinnaCortes-1)) are [supervised learning](http://en.wikipedia.org/wiki/Supervised_learning) models with associated learning [algorithms](http://en.wikipedia.org/wiki/Algorithm) that analyze data and recognize patterns, used for [classification](http://en.wikipedia.org/wiki/Statistical_classification) and [regression analysis](http://en.wikipedia.org/wiki/Regression_analysis). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-[probabilistic](http://en.wikipedia.org/wiki/Probabilistic_classification) [binary](http://en.wikipedia.org/wiki/Binary_classifier) [linear classifier](http://en.wikipedia.org/wiki/Linear_classifier). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Follow the instructions in [1] to examine **SMO** and **LibSVM**, and fill in the following table:

|  |  |  |
| --- | --- | --- |
| **Dataset** | **SMO’s classifier model and performance** | **LibSVM’s classifier model and performance** |
| diabetes |  |  |

Notice: A wrapper class for the libsvm tools (the libsvm classes, typically the jar file, need to be in the classpath to use this classifier) >> see <http://weka.wikispaces.com/LibSVM>

1. http://www.cs.waikato.ac.nz/ml/weka/mooc/dataminingwithweka/ [↑](#footnote-ref-1)