CS 539 Machine Learning

Project 6 - Instance-Based Learning and Regression Methods

Chris Winsor

The Spambase Dataset

The Spambase Dataset consists of e-mail attributes like word and character frequency, and a class variable which identifies whether the mail is spam or not. It has 4601 samples, 57 attributes and one class variable. The attributes are numeric, the class variable is nominal.

Guiding Questions

- Using just the attributes in the dataset - can IB1, IBk, Linear Regression, and Locally Weighted Linear Regression (LWLR) predict whether the e-mail is spam?
- How does the size of the dataset affect runtime ? How does the 'K' in IBk affect the total runtime ?

Preprocessing the Spambase Dataset

Attributes

The original dataset has 57 attributes. They are numeric values, unbinned. There is a lot of cross-correlation between attributes.

Class Variable

There is one class variable which is nominal.

Preprocessing Steps:

- For IB1 and IBk there is no preprocessing. These techniques can predict a nominal class variable using numeric attributes. To test performance we split the dataset into 100, 500, and 1000 entity subsets using Weka
- For Linear Regression and LWLR the class variable is converted to a numeric value (0.0 and 1.0) because these techniques require a numeric class variable.

Performance Metrics

Accuracy: measured as % correct for nominal class and mean-squared error for numeric class.
Total Runtime: calculated as Training + Test (Lazy techniques have a training time of zero).

Experiment 1 - IB1, IBX with Nominal Target Class

In this experiment we investigate the Spambase dataset using IB1 and IBX. In this case the target "spam" class variable is a nominal value.

We first establish baselines using Zero-R, One-R, J4.8 and Neural Network. We then run IB1 and IBX with various values of K and various number of instances in the dataset. Settings for all classifiers were the default except Neural Network which was run with 3 hidden nodes.

	100 instances		1000 instances		4601 instances	
	Percent Correct	Total Time (sec)	Percent Correct	Total Time (sec)	Percent Correct	Total Time (sec)
ZeroR	50.00	0.00	60.72	0.00	60.60	0.00
OneR	78.10	0.00	77.30	0.01	78.48	0.06
J4.8	80.50	0.01	89.27	0.15	92.68	0.95
Neural Net	80.20	0.41	90.43	3.74	91.71	16.71
IBK (K=1)	83.00	0.00	87.74	0.04	90.75	0.65
IBK (K=2)	85.40	0.00	85.52	0.04	88.92	0.74
IBK (K=5)	78.60	0.00	86.10	0.05	90.11	0.84
IBK (K=7)	78.00	0.00	85.34	0.05	89.87	0.88
IBK (K=10)	78.00	0.00	85.57	0.06	89.62	0.97
IBK (K=20)	70.90	0.00	84.86	0.07	88.59	1.06
IBK (K=50)	73.60	0.00	83.13	0.08	86.77	1.24

Summary

- IBK's speed (Total Time) was surprisingly good. In the most strenuous case (K=50 and 4601 instances) the total time for IBK was 1.24 seconds as compared to Neural Net of 16.71 seconds. IBK took about 30% more time than J7.4.
- IBK's accuracy (Percent Correct) was on par with both J4.8 and Neural Net.

Experiment 2 - Linear Regression and LWLR with Numeric Target Class

In this experiment we investigate the Spambase dataset using Linear Regression and Locally Weighted Linear Regression. In this case the "spam" class variable is a numeric value.

We first establish baselines using ZeroR and Neural Network. We then run Linear Regression and Locally Weighted Linear Regression with a variety of dataset sizes. Settings for all classifiers were the default except Neural Network which was run with 3 hidden nodes.

	100 Instances		1000 Instances		4601 Instances	
	Root-mean-squared Error	Total Time (sec)	Root-mean-squared Error	Total Time (sec)	Root-mean-squared Error	Total Time (sec)
Zero-R	0.50	0.00	0.49	0.00	0.49	0.00
Neural Net	0.39	0.26	0.33	2.67	0.32	12.37
Linear Regression	0.87	0.01	0.35	0.10	0.34	0.26
LWL (K=1)	0.45	0.00	0.33	0.04	0.30	0.54
LWL (K=2)	0.45	0.00	0.33	0.07	289.35	0.66
LWL (K=5)	0.45	0.00	0.69	0.09	0.99	0.80
LWL (K=10)	0.52	0.01	0.53	0.11	1.29	0.96
LWL (K=20)	0.95	0.01	2.45	0.17	1.75	1.19
LWL (K=50)	6.28	0.06	2.03	0.49	1.69	2.48

Summary

LWL's speed (Total Time) was reasonable. In the most strenuous case (K=50 and 4601 instances) LVL took 2.48 seconds as compared to Neural Net's time of 12.37 seconds. LWL was about 10 times slower than Linear Regression.

LWL's accuracy (Root-Mean-Squared Error) was on par with its peers. Interestingly the RMS error increased with K, and this is not explained. In addition, there is an anomaly in RMS, specifically for LWL (K=2) with the 4601 instances the RMS error was way above anything in its peer group.