Chapter 9

ROC Analysis for Single and Aggregate Models

Recall from section 4.7 that a discrete classifier simply assigns a class label to a test instance (Fawcett, 2001, 2004, 2006).The single, OVA aggregate and pVn aggregate models were treated as discrete classifiers for the predictive performance analysis reported in chapters 7 and 8. Even though the single, OVA aggregate and pVn aggregate models assign probabilistic scores to the test instances as discussed in section 6.4, the scores could not be used in the statistical tests used in chapters 7 and 8 to compare model performance. Student's paired samples t-test, the Diff(A,S) measure, and the Ratio(A,S) measures that were used to compare model performance do not provide the capability for the analysis of the probabilistic scores assigned to the model predictions.

Receiver Operating Characteristic (ROC) curves and ROC analysis were discussed in section 4.7. ROC analysis enables the analysis of classifiers based on the scores that are assigned to the test instances. The classification models of chapters 7 and 8 were treated as probabilistic classifiers for the ROC analysis reported in this chapter. The purpose of the ROC analysis was to answer the questions below in order to establish whether the aggregate models provide a better level of performance compared to the single models for different operating conditions:

1. Do OVA aggregate models provide a higher level of predictive performance compared to single models for different operating conditions?

2. Do pVn aggregate models provide a higher level of predictive performance compared to single models for different operating conditions?

This chapter is organised as follows: Sections 9.1 and 9.2 respectively provide a discussion of 2-class and multi-class ROC analysis. Section 9.3 provides a discussion of ROC analysis for the 5NN single and aggregate models. Section 9.4 provides a discussion of ROC analysis for the See5 single and aggregate models. Section 9.5 concludes the chapter.

9.1 ROC analysis for 2-class predictive models

Recall that ROC curves provided a graphic representation of predictive model performance for 2-class prediction tasks (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000). A probabilistic classification model typically assigns a class and a score for the class. Most commonly, the score is the probability that a test instance belongs to the predicted class (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000). ROC analysis is concerned with the selection of the model with the optimal performance based on the cut-off point (threshold) λ that is used to decide when an instance should be declared positive or negative. A cut-off point (threshold) is the score value $conf(\mathbf{x})$ for which $conf(\mathbf{x}) \geq \lambda$ implies that the predicted class for instance $\mathbf{x}$ is the positive class. ROC analysis may also be used to determine which of two models provides a higher level of predictive performance as discussed in section 4.7. ROC analysis produces a statistic called the Area Under ROC curve (AUC). Recall from section 4.7.3 that when the predictive performance of a probabilistic classifier is better than random guessing then $AUC = AUC_{below} + AUC_{above}$. AUC_{below} and AUC_{above} are respectively the area below and the area above the 45 degree line which represents random guessing in the 2-dimensional ROC plane. The AUC is also the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). Given two probabilistic classifiers, the classifier with the larger AUC value has a higher level of predictive performance under different operating conditions.

9.2 ROC analysis for multi-class predictive models

Computation of the AUC for 2-class classification models is a straight forward task. ROC analysis for *k*-class ($k > 2$) prediction tasks is concerned with the Volume Under the ROC Surface (VUS). Computation and visualisation of the VUS is a non-trivial task. Fawcett (2004, 2006) has discussed two approximations of the VUS measure that have been proposed by Hand and Till (2001) and Provost and Domingos (2001). The Hand and Till (2001) measure is defined as

$$AUC_{total} = \frac{2}{k.(k-1)} \sum_{i,j \le k, i \ne j}^{k} AUC(c_i, c_j) \tag{9.1}$$

where $AUC(c_i, c_j)$ is the AUC value computed for the two class ROC curve for the classes c_i and c_j and *k* is the number of classes for the predictive model.

The Provost and Domingos (2001: cited by Fawcett, 2004, 2006) measure is defined as

$$AUC_{total} = \sum_{i=1}^{k} P_r(c_i).AUC(c_i, rest) \tag{9.2}$$

where $AUC(c_i, rest)$ is the *AUC* for class c_i compared to all the other *k*-1 classes and $P_r(c_i)$ is the prevalence (prior probability) of class c_i in the training dataset(s). The Provost and Domingos (2001) measure is commonly called the *one-versus-rest* approximation of the VUS (Fawcett, 2001, 2004, 2006). The Provost and Domingos (2001) measure is easier to visualise and faster to compute. However, determining the prevalence (prior probability) $P_r(c_i)$ of a class is a simple matter for a single predictive model. When base models are based on boosted training datasets and then combined into one aggregate model, the determination of $P_r(c_i)$ is not straight forward any more. A modified version of the Provost and Domingos (2001) measure that was designed by the author of this thesis and used for the ROC analysis of this chapter is a simple mean value for the AUC and is defined as

$$AUC_{total} = \frac{1}{k} \sum_{i=1}^{k} AUC(c_i, rest) \tag{9.3}$$

where $AUC(c_i, rest)$ has the same meaning as before and *k* is the number of classes for the multi-class (*k*-class) prediction task. The justification for computing the mean value of $AUC(c_i, rest)$ in equation (9.3) is as follows: The VUS estimates of equations (9.1) and (9.2) are based on the arithmetic combination of the AUC values for many 2-dimensional planes in multi-class ROC space. Equation (9.1) computes a mean value for $k(k-1)/2$ such planes. Equation (9.2) computes a simple sum of weighted values of the AUC for *k* 2-dimensional planes. Given the foregoing

observations, computation of the mean AUC in equation (9.3) gives a useful estimate of the VUS, especially for purposes of comparing the performance of two multi-class probabilistic classifiers.

Several values need to be computed in order to derive the approximation of the VUS. The values that were computed and the methods for computation of these values are given in table 9.1. Since 10 test sets were used to measure model performance, it was necessary to combine the test results for the TPRATE and FPRATE into summary measures for the 10 test sets. The mean TPRATE and mean FPRATE were computed for each threshold value for the probabilistic classifier. Fawcett (2004, 2005) calls this approach *threshold averaging*.

Table 9.1: Computations for the estimation of the VUS

Value	Description	Computation
Mean $TPRATE(c_i, rest, \lambda)$	Mean TPRATE for probabilistic classifier $PC(c_i, rest)$ for threshold value λ	Mean values computed using 10 test sets
Mean $FPRATE(c_i, rest, \lambda)$	Mean FPRATE for probabilistic classifier $PC(c_i, rest)$ for threshold value λ	Mean values computed using 10 test sets
$AUC(c_i, rest)$	AUC computed for the curve defined by the mean TPRATE and FPRATE values for probabilistic classifier $PC(c_i, rest)$ for different λ values.	Integration of the area between the curve and the 45^0 line in the 2-dimensional ROC space. The λ values for the 5NN probabilistic classifiers were: 0.6, 0.8 and 1.0. The values for See5 were: 0.5, 0.75 and 1.0.
Mean AUC_{total}	Estimation of VUS	Computed using equation (9.3)

9.3 ROC analysis for 5NN models

The ROC analysis results for the 5NN single and aggregate models for the forest cover type, KDD Cup 1999 and wine quality datasets are given in table 9.2. The details of the ROC analysis are given in the appendix tables G.2, G.3 and G.4. The AUC_{above} values and *Gini* concentration coefficients for the probabilistic classifiers are given in table 9.2 columns 3 to 10 for each class. The mean AUC_{above} and mean *Gini* values for the single *k*-class model and aggregate *k*-class models are also given in the table. Recall from sections 4.7.3 and 9.1 that AUC_{above} is the area between the ROC curve and the 45 degree line and $Gini = 2\text{x}AUC_{above}$. When the 2-

dimensional ROC space is visualised as a grid of 100 cells with each cell having a width of 0.1 and a height of 0.1, then an increment of 0.01 in the AUC corresponds to an AUC increase of one such cell. This corresponds to a 2% increase in the area AUC_{above} whose maximum value is 0.5, and an increase of 4% in the *Gini* concentration coefficient whose maximum value is 1.0.

Table 9.2: ROC analysis results for the 5NN single and aggregate models

Dataset, algorithm	Probabilistic classifier *PC(c$_i$,rest)*	*AUC$_{above}$* and *Gini* concentration coefficient for model:							
		single		un-boosted OVA		boosted OVA		pVn	
		AUC$_{above}$	*Gini*	*AUC$_{above}$*	*Gini*	*AUC$_{above}$*	*Gini*	*AUC$_{above}$*	*Gini*
Forest cover type, 5NN	PC(1,rest)	0.29	0.58	0.33	0.66	0.33	0.66	0.32	0.64
	PC(2,rest)	0.23	0.46	0.28	0.56	0.30	0.60	0.27	0.54
	PC(3,rest)	0.25	0.50	0.35	0.70	0.34	0.68	0.31	0.62
	PC(4,rest)	0.45	0.90	0.44	0.88	0.49	0.98	0.48	0.96
	PC(5,rest)	0.43	0.86	0.46	0.92	0.47	0.94	0.46	0.92
	PC(6,rest)	0.33	0.66	0.38	0.76	0.37	0.74	0.36	0.72
	PC(7,rest)	0.47	0.94	0.47	0.94	0.47	0.94	0.46	0.92
	Mean	**0.35**	**0.70**	**0.39**	**0.78**	**0.40**	**0.80**	**0.38**	**0.76**
KDD Cup 1999, 5NN	PC(NORMAL,rest)	0.36	0.72	0.41	0.82	0.41	0.82	0.43	0.86
	PC(DOS,rest)	0.33	0.66	0.33	0.66	0.33	0.66	0.48	0.96
	PC(PROBE,rest)	0.44	0.88	0.44	0.88	0.44	0.88	0.49	0.98
	PC(R2L,rest)	0.30	0.60	0.29	0.58	0.27	0.54	0.38	0.76
	PC(U2R,rest)	0.15	0.30	0.21	0.42	0.20	0.40	0.13	0.26
	Mean	**0.32**	**0.64**	**0.33**	**0.66**	**0.33**	**0.66**	**0.38**	**0.76**
	PC(4,rest)	0.04	0.08	0.04	0.08	0.04	0.08	0.03	0.06
	PC(5,rest)	0.15	0.30	0.17	0.34	0.18	0.36	0.16	0.32
	PC(6,rest)	0.03	0.06	0.04	0.08	0.06	0.12	0.12	0.24
	PC(7,rest)	0.09	0.18	0.11	0.22	0.12	0.24	0.10	0.20
	PC(8,rest)	0.04	0.08	0.04	0.08	0.04	0.08	0.05	0.10
	Mean	**0.07**	**0.14**	**0.08**	**0.16**	**0.09**	**0.18**	**0.09**	**0.18**

The mean AUC_{above} values for the forest cover type models range between 0.35 and 0.40. The boosted OVA aggregate model provided the best performance (0.40), followed by the un-boosted OVA aggregate model (0.39) followed by the pVn model (0.38). Since the single model has a mean AUC_{above} of 0.35, all forest cover type aggregate models provided an increased level of predictive performance over the single model. An examination of the performance on the individual classes reveals that the aggregate models provided increased performance levels on six out of the seven classes. There were no improvements on class 7.

The mean AUC_{above} values for the KDD Cup 1999 models range between 0.32 and 0.38. The pVn aggregate model provided the best performance (0.38), followed by the un-boosted and boosted OVA aggregate models (0.33). Since the single model has a mean AUC_{above} of 0.32, the OVA models provided a very slight improvement in predictive performance. The pVn model provided a much higher performance improvement over the single model. The AUC_{above} values for the individual classes indicate that the KDD Cup 1999 pVn aggregate model provided increased performance levels on four out of the five classes. The un-boosted and boosted OVA aggregate models each provided increased performance levels on two out of five classes.

The mean AUC_{above} values for the wine quality models are very small. The values range between 0.07 and 0.09. The boosted OVA and pVn aggregate models provided the best performance (0.09), followed by the un-boosted OVA aggregate models (0.08). Since the single model has a mean AUC_{above} of 0.07, the OVA and pVn models provided a slight improvement in predictive performance. The AUC_{above} values for the individual classes indicate that the wine quality pVn aggregate model provided increased performance levels on four out of the five classes. The un-boosted and boosted OVA aggregate models each provided increased performance levels on three out of five classes.

9.4 ROC analysis for See5 models

The ROC analysis results for the See5 single and aggregate models for the forest cover type, KDD Cup 1999, and wine quality datasets are given in table 9.3. The details of ROC analysis are given in the appendix tables G.5, G.6 and G.7. The AUC_{above} values for the probabilistic classifiers are given in table 9.2 columns 3 to 10 for each class. The mean AUC_{above} and mean *Gini* values for the single *k*-class model and aggregate *k*-class models are also given in the table.

The mean AUC_{above} values for the See5 forest cover type models range between 0.36 and 0.38. The boosted OVA and pVn aggregate models provided the best performance (0.38), followed by the single model (0.37) followed by the un-boosted

OVA aggregate model (0.36). Since the single model has a mean AUC_{above} of 0.37, the boosted OVA and pVn aggregate models for forest cover type provided an increased level of predictive performance over the single model. The un-boosted OVA aggregate model did not provide any performance gains. An examination of the performance on the individual classes reveals that the boosted OVA aggregate model provided increased performance on five out of the seven classes. The pVn aggregate model provided increased performance on six out of the seven classes.

Table 9.3: ROC analysis results for the See5 single and aggregate models

Dataset, algorithm	Probabilistic classifier *PC(ci,rest)*	*AUCabove* and *Gini* concentration coefficient for model:							
		single		un-boosted OVA		boosted OVA		pVn	
		AUCabove	*Gini*	*AUCabove*	*Gini*	*AUCabove*	*Gini*	*AUCabove*	*Gini*
Forest cover type, See5	PC(1,rest)	0.27	0.54	0.28	0.56	0.30	0.60	0.31	0.62
	PC(2,rest)	0.29	0.58	0.22	0.44	0.30	0.60	0.30	0.60
	PC(3,rest)	0.29	0.58	0.30	0.60	0.30	0.60	0.34	0.68
	PC(4,rest)	0.46	0.92	0.43	0.86	0.47	0.94	0.47	0.94
	PC(5,rest)	0.42	0.84	0.45	0.90	0.42	0.84	0.43	0.86
	PC(6,rest)	0.37	0.74	0.36	0.72	0.36	0.72	0.39	0.78
	PC(7,rest)	0.47	0.94	0.45	0.90	0.48	0.96	0.45	0.90
	Mean	**0.37**	**0.74**	**0.36**	**0.72**	**0.38**	**0.76**	**0.38**	**0.76**
KDD Cup 1999, See5	PC(NORMAL,rest)	0.38	0.76	0.44	0.88	0.41	0.82	0.40	0.80
	PC(DOS,rest)	0.40	0.80	0.25	0.50	0.27	0.54	0.34	0.68
	PC(PROBE,rest)	0.17	0.34	0.39	0.78	0.41	0.82	0.48	0.96
	PC(R2L,rest)	0.18	0.36	0.12	0.24	0.11	0.22	0.26	0.52
	PC(U2R,rest)	0.31	0.62	0.23	0.46	0.19	0.38	0.38	0.76
	Mean	**0.29**	**0.58**	**0.29**	**0.58**	**0.28**	**0.56**	**0.37**	**0.74**
Wine quality white, See5	PC(4,rest)	0.11	0.22	0.16	0.32	0.16	0.32	0.14	0.28
	PC(5,rest)	0.18	0.36	0.17	0.34	0.18	0.36	0.19	0.38
	PC(6,rest)	0.05	0.10	0.01	0.02	0.01	0.02	0.11	0.22
	PC(7,rest)	0.14	0.28	0.09	0.18	0.10	0.20	0.16	0.32
	PC(8,rest)	0.04	0.08	0.05	0.10	0.06	0.12	0.06	0.12
	Mean	**0.10**	**0.20**	**0.10**	**0.20**	**0.10**	**0.20**	**0.13**	**0.26**

The mean AUC_{above} values for the See5 KDD Cup 1999 models range between 0.29 and 0.37. The pVn aggregate models provided the best performance (0.37), followed by the single model and un-boosted OVA aggregate model (0.29) followed by the boosted OVA aggregate model (0.28). Since the single model has a mean AUC_{above} of 0.29, the pVn aggregate models for KDD Cup 1999 provided an increased level of

predictive performance over the single model. The OVA aggregate models did not provide any performance gains. An examination of the performance on the individual classes reveals that the pVn aggregate model provided increased performance on four out of the five classes.

The mean AUC_{above} values for the See5 wine quality models are very small. The single, un-boosted OVA, and boosted OVA models have values of 0.10 for the mean AUC_{above}. These results indicate that the OVA aggregate models did not provide any performance gains. The pVn aggregate model provided the best performance with a mean AUC_{above} value of 0.13 which indicates an increased level of predictive performance over the single model. An examination of the performance on the individual classes reveals that the pVn aggregate model provided increased performance on all five classes.

9.5 Conclusions

The single and aggregate models of chapters 7 and 8 were treated as probabilistic classifiers for the ROC analysis discussed in this chapter. The first question that was posed for this chapter was: *Do OVA aggregate models provide a higher level of predictive performance compared to single models for different operating conditions?* Performance improvements were realised for the 5NN OVA aggregate models and the See5 boosted aggregate model for the forest cover type dataset. No performance gains were realised for the See5 un-boosted OVA aggregate model. No performance gains were realised from the OVA aggregate models for the 5NN and See5 algorithms for the KDD Cup 1999 and wine quality datasets.

The conclusion from the foregoing observations is that OVA aggregate modeling as proposed in this thesis may or may not result in improved performance. Schaffer (1994) has observed that no single strategy for machine learning is better at generalisation (prediction) than all other strategies for all problem domains. The above conclusion should therefore be viewed in the context of Schaffer's (1994) observation. The single model confusion matrices of the forest cover type 5NN and See5 models had higher levels of sparsity compared to the KDD Cup and wine quality single models. It can be concluded that OVA modeling, as proposed in this

thesis, provides performance improvements for a dataset whose confusion matrix has a high level of sparsity.

The second question that was posed for this chapter was: *Do pVn aggregate models provide a higher level of predictive performance compared to single models for different operating conditions?* The pVn aggregate models provided performance improvements for the forest cover type, KDD Cup1999, and wine quality datasets for both the 5NN and See5 algorithms. It can be concluded that pVn modeling provides performance improvements as long as the single model for a dataset has the sparse confusion matrix property.

In conclusion, the observations based on the ROC analysis of this chapter support the conclusions of chapter 7 and 8. The ROC analysis results have additionally demonstrated that OVA and pVn aggregate models can provide better predictive performance under different operating conditions compared to single models. Based on the conclusions of chapters 5, 7, 8 and this chapter, recommendations are given in the next chapter for dataset selection and aggregate modeling from large datasets.

Chapter 10
Recommendations for Dataset Selection

> '…the problems in science … on a deeper level ... are directed towards a consensus, or rational agreement, between the parties concerned …' (Toulmin et al, 1979)

The studies conducted on feature selection, training dataset selection, and aggregate modeling, the experimental results and analysis of the results were presented in chapters 5 to 9. This chapter provides an integrated discussion of the experimental results by giving a summary of the results. The chapter also provides theoretical models that were derived from the results and suggestions on how to conduct feature and training dataset selection for aggregate modeling from large datasets. Recall from section 4.3.5 that several researchers have argued for the need for empirically derived theories for computer systems (Simon, 1996), machine learning (Dietterich, 1997) and artificial intelligence systems (Cohen, 1995). It is the author's opinion that empirically derived theoretical models for data mining should provide value for researchers and practitioners in data mining. Recall that the main research question for the thesis was:

What methods of dataset selection can be used to obtain as much information as possible from large datasets while at the same time using training datasets of small sizes to create predictive models that have a high level of predictive performance?

The following sections provide several concise answers to this question. A summary of the methods that were used for the reduction of prediction error is given in section 10.1. Theoretical models and recommendations for feature selection and training dataset selection are provided in sections 10.2 and 10.3 respectively. Section 10.4 provides a summary of the chapter.

10.1 Reduction of prediction error

It was argued in chapter 2 that a high level of predictive performance should be achieved when training datasets are selected with the main objective of reducing prediction error. Chapter 2 provided a discussion of the components that make up

the predictive error, namely bias, variance and intrinsic error. The methods that have the potential to reduce the bias and variance error components were discussed in chapter 2. The use of simple models (Dietterich & Bakiri, 1995) and boosting (Freund & Schapire, 1997) are known to reduce the bias component of the prediction error. The use of aggregate modeling (Breiman, 1996) and the use of simple models (e.g. OVA base models) are known to reduce variance error. The use of good feature subsets for prediction (Dietterich & Kong (2005) and reduction of noise through sampling (Smyth, 2001) are known to reduce the variance error.

The main objective of the experiments reported in chapters 5, 7 and 8 was to reduce the bias and variance components of prediction error using the methods stated above. This was achieved through:
(1) The use of many (relatively) small samples for correlation measurement and base model construction.
(2) The design of simple base models, each of which specialises in the prediction of a subset of the k classes ($k > 2$) for the prediction task and uses a different training dataset from the other base models.
(3) The design of training datasets for base models, with the objective of increasing the coverage of those regions of the instance space where correct prediction is more difficult.

10.2 Recommendations for feature selection

This section provides a summary of the discussion of the studies that were conducted for feature selection as reported in chapter 5. A theoretical model of the factors that affect the quality of selected features is proposed and guidelines are provided on how to proceed with feature selection in the presence of large datasets. Section 10.2.1 provides a summary of the feature selection studies. Section 10.2.2 provides guidelines for feature selection based on the reported experimental results.

10.2.1 Summary of the feature selection experimental results

The factors that affect the quality of selected features for single models were discussed in sections 5.6 and 5.7. In the context of this discussion, quality refers to the extent to which as many relevant features as possible are included, and as many

irrelevant and redundant features as possible are excluded from the selected subset of predictive features. The point was made in chapter 3 that existing literature in computational data mining indicates that most commonly a single sample of (all available) data is used to measure class-feature and feature-feature correlations for feature ranking. Probes (fake variables) have been used for the validation of class-feature correlations. The experiments of chapter 5 demonstrated that class-feature correlations measured from samples of a large dataset can vary widely from sample to sample. The point was also made in chapter 3 that in computational data mining, mathematical functions are commonly used as heuristic measures by feature subset search algorithms. The experimental results of chapter 5 revealed that the use of mathematical functions as heuristic measures does not always result in the best decisions for the features to be included in the subset of the best predictive features.

Based on the experimental results and conclusions of chapter 5, the following are research contributions of this thesis to the problem of feature selection:
(1) Reliable methods of measuring class-feature and feature-feature correlations through the use of many samples.
(2) Reliable feature ranking through the use of mean class-feature correlations values.
(3) Reliable class-feature and feature-feature correlation validation through the use of mean values for the class-probe correlations to eliminate non-relevant features.
(4) Usage of decision rules for heuristics evaluation of the best feature to select at a given decision point for a feature subset search algorithm.

Arising from the discussions of chapter 3 and the experimental results of chapter 5, the theoretical model shown in figure 10.1 was developed for purposes of representing the relationships between the factors that have an effect on the quality of selected features for predictive classification modeling. The theoretical model of figure 10.1 offers a predictive theory of the outcome of feature selection as depicted in figure 4.2 of section 4.3.3. However, the model of figure 10.1 does not provide causal explanations as depicted in figure 4.2. Proper causation experiments, with experiment controls are needed in order to conclude beyond reasonable doubt that the relationships shown in figure 10.1 are due to the indicated factors and not fully or partially due to other factors (Cohen, 1995: ch.9).

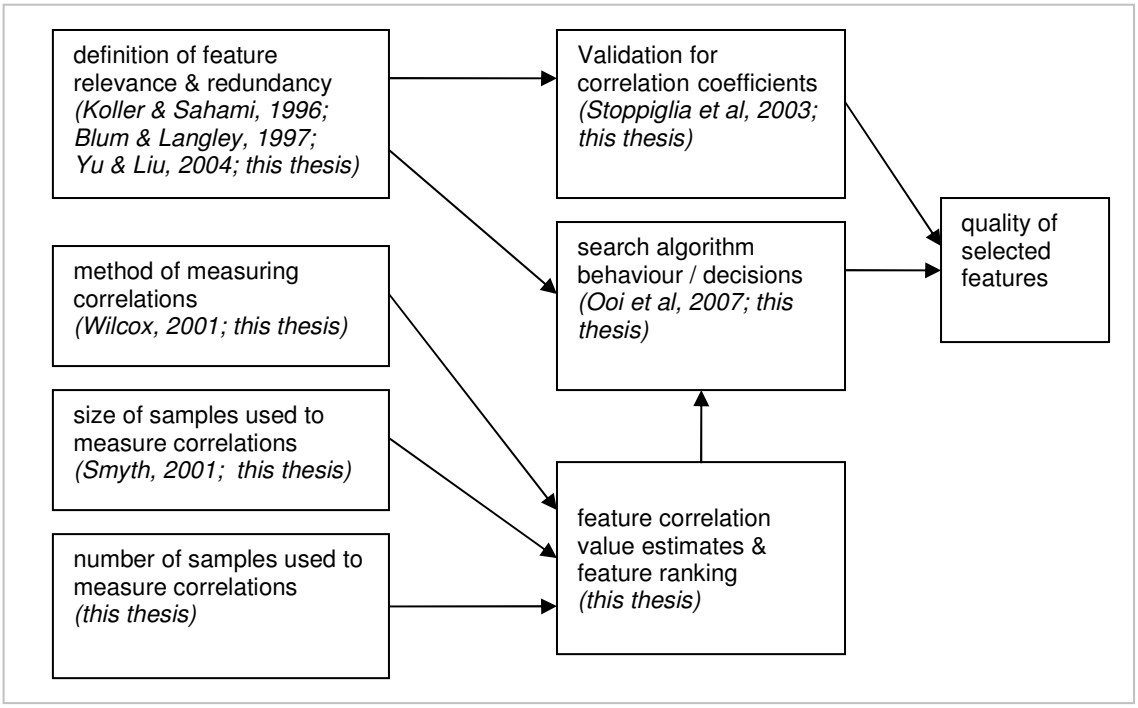

Figure 10.1: Theoretical predictive model for feature selection using filtering methods

The theoretical model of figure 10.1 predicts that the definitions of feature relevance and redundancy that are used in the procedures for the validation of the class-feature correlation coefficient estimate, will affect the outcome of the validation procedure. When feature ranking is all that is required, then the quality of the selected features will be affected by the outcome of the validation procedure. The definitions of feature relevance and redundancy will affect the behaviour of a feature subset search algorithm in terms of the final subset of selected features. The method used to measure the correlation coefficients, the sample sizes used, and the number of samples used, will all affect the estimate of the correlation coefficients, and in turn affect the feature ranking that is generated for input to the search algorithm. Finally, the quality of the feature subset selected by a search algorithm is influenced by the quality of the decisions made by the search algorithm.

10.2.2 Guidelines for feature selection

The steps shown in figure 10.2 are recommended for feature selection from large datasets of moderate dimensionality.

Step 1: Obtain information on the meaning of *low, medium* and *high* correlation for the domain from where the data originates. If this information is not available, use Cohen's (1988) guidelines.

Step 2: Take many small random samples and add one or more probes to each sample. Ten test samples of 1000 instances and at least one probe (Gaussian for quantitative continuous data, Uniform for quantitative discrete and qualitative data) provided useful information for the chapter 5 experiments.

Step 3: Measure the class-probe, class-feature and feature-feature correlations using a robust measure of association, eg. Kendall's tau or Pearson's r with the outliers removed.

Step 4: Compute the mean class-probe, class-feature and feature-feature correlations. If the confidence intervals of the means for the correlation values are large, go back to step 1 and increase the sample size.

Step 5: Conduct feature ranking based on the mean values of the class-probe and class-feature correlations.

Step 6: Use the probe method discussed in chapter 5 to eliminate all features whose ranking is below that of any of the probes from further consideration, as discussed in chapter 5.

Step 7: If the feature selection task is to select a pre-defined number of features (*w*), then select the top *w* features that have a correlation coefficient of practical significance for the problem domain and stop. Alternatively, a user-specified threshold for correlation values can be used to determine which features to select.

Step 8: If the feature selection task is to identify the best subset of features then construct decision rules for the meanings of relevance and redundancy for the problem domain where the dataset originates. If this information is not available, use Cohen's (1988) guidelines.

Step 9: Conduct the feature subset search using the decision rules of step 8 to obtain the best feature subset.

Figure 10.2: Recommended procedure for feature selection from large datasets

If the feature selection task is to select a pre-specified number of features, then steps 1 to 7 of figure 10.2 are recommended. If on the other hand, the task is to select the best subset of features, then steps 1 to 7 should be followed by steps 8 and 9. Step 9 involves the search for the best feature subset. Suggestions on how to conduct steps 1 to 7 using commonly available software (SPSS and MS Excel) are given in the appendix table H.1. The decision-rule base feature selection algorithm that was presented in chapter 5 is a good candidate for performing step 9. One alternative to

the above approach is to conduct steps 1 to 7 of figure 10.2 followed by Yu and Liu's (2004) method of redundancy analysis that was discussed in chapter 3.

10.3 Recommendations for training dataset selection for aggregate modeling

This section provides a summarised discussion of the studies that were conducted for OVA and pVn base model design and training dataset selection as well as the implications of the experimental results. A theoretical model that was developed for the factors that affect the quality of selected training datasets, based on existing literature is presented. An extension of the theoretical model based on the studies conducted for this thesis is proposed, and guidelines are provided on how to proceed with training dataset selection for aggregate model implementation in the presence of large datasets. Section 10.3.1 provides a summary of the training dataset selection experiments and the research contributions arising from the experiments. Section 10.3.2 presents the theoretical model for training dataset selection. Parallel and serial aggregation methods are discussed in section 10.3.3. Guidelines for training dataset selection are provided in section 10.3.4.

10.3.1 Summary of the training dataset selection experimental results

Sections 7.4 and 7.5 provided the discussion and conclusions for the OVA model dataset selection experiments. Sections 8.5 and 8.6 provided the discussion and conclusions for the pVn model dataset selection experiments. Chapter 9 provided the results for ROC analysis to compare single models, OVA and pVn aggregate models. The main conclusions from chapters 7, 8 and 9 were that the proposed dataset selection methods for OVA and pVn aggregate modeling generally provided improvements in predictive performance. In summary, the main research contributions arising from the reported experiments are as follows:

(1) The use of OVA modeling to increase the amount of training data used for modeling from large datasets, and to increase the level of predictive performance.

(2) The use of pVn modeling to increase the amount of training data used for modeling from large datasets, and to increase the level of predictive performance. pVn modeling reduces the number of base models compared to OVA modeling.

(3) The use of a confusion matrix to provide information for the design of boosted OVA and pVn base models.
(4) The use of a confusion graph as a graphical and mathematical representation of the information in a confusion matrix to be used as input to the algorithm for determining the positive and negative classes of pVn base models.
(5) A definition of the sparse confusion matrix property which can be used to determine whether boosted OVA and pVn base models will provide performance improvements for a given dataset.
(6) A base model combination algorithm for KNN OVA and pVn base model predictions. The algorithm resolves tied predictions.

10.3.2 Theoretical model for training dataset selection

A theoretical model to summarise the work on aggregate modeling as reported in chapter 2 was developed by the author. The theoretical model is shown in figure 10.3. One major factor that affects the performance of aggregate models is syntactic diversity. Recall from chapter 2 that the term syntactic diversity refers to the level of dis-similarity between the base models that make up an aggregate model. Syntactic diversity has been achieved by researchers (as indicated in figure 10.3) either through variation of the learning task, or variation of the base model structure, or variation of the training datasets for base models. A second major factor that affects aggregate model performance is the predictive accuracy of the base models. Several researchers (as indicated in figure 10.3) have achieved a high level of base model predictive accuracy (Chan & Stolfo, 1998) or single model accuracy (Kubat & Matwin, 1997) through sampling methods that balance the level of class representation for datasets with skewed class distributions. A second approach has been to vary the learning task and/or the base model structure.

Syntactic diversity, predictive accuracy of the base models and the method of determining the winning class lead to a reduction of the bias and variance components of the prediction error of an aggregate model as depicted in figure 10.3. The level to which the bias and variance components of the prediction error are reduced affects the predictive performance of the aggregate model.

The research for this thesis concentrated on the selection of training datasets from large amounts of data, with the objective of constructing aggregate models which

provide a high level of predictive performance. The methods of training dataset selection that were studied were aimed at achieving variation in the base model structures, variation in the training datasets for the base models, and balancing of the class representation in the base models. The theoretical model shown in figure 10.4 is an extension of the model of figure 10.3, based on the studies conducted for this thesis.

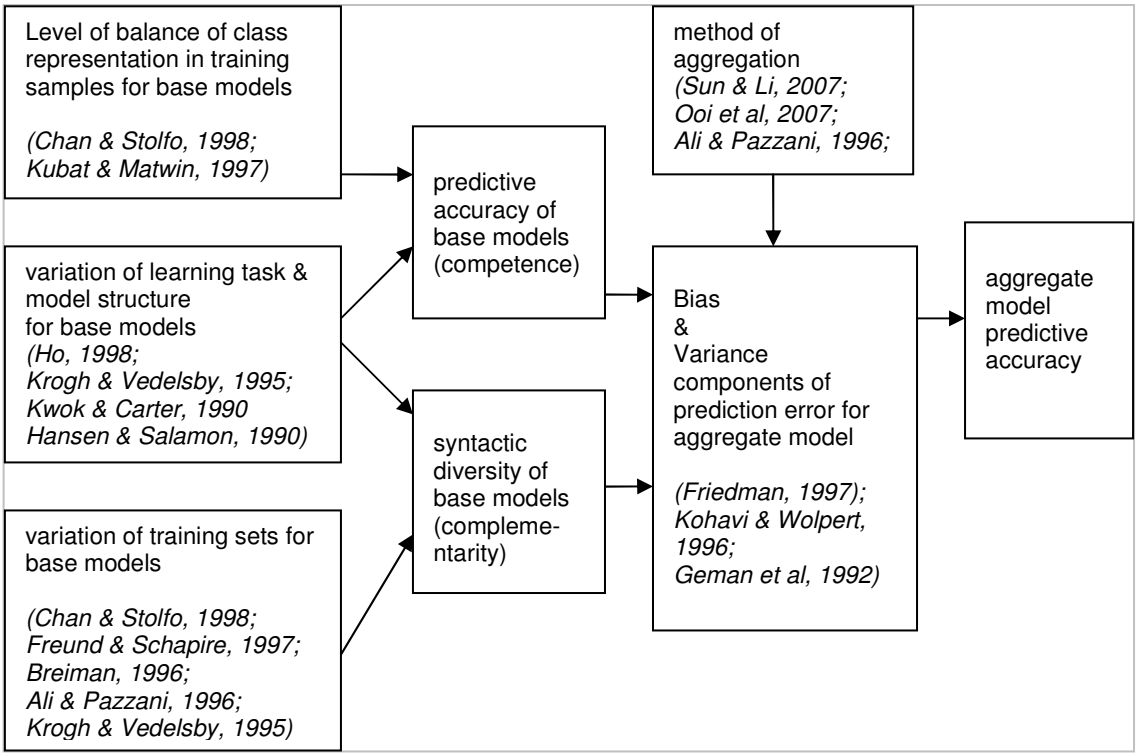

Figure 10.3: Theoretical predictive model for aggregate model performance based on existing literature

The model of figure 10.4 predicts that the use of information about the structure of the instance space combined with information on the aggregation method should result in the design of a set of base models whose performance should ultimately result in high predictive performance. The design of the base models should influence the methods used to select the training sample for each base model from the large dataset. The methods of training dataset selection, based on the designed base models, should influence the level of balance of the classes in the training datasets, the level of variation in the training datasets for the base models, and the level of variation in the learning tasks and structures of the base models. This should in turn influence the predictive performance of each base model, and the syntactic diversity in the set of base models. The algorithm used for combining the base model predictions will affect the bias error of the aggregate model.

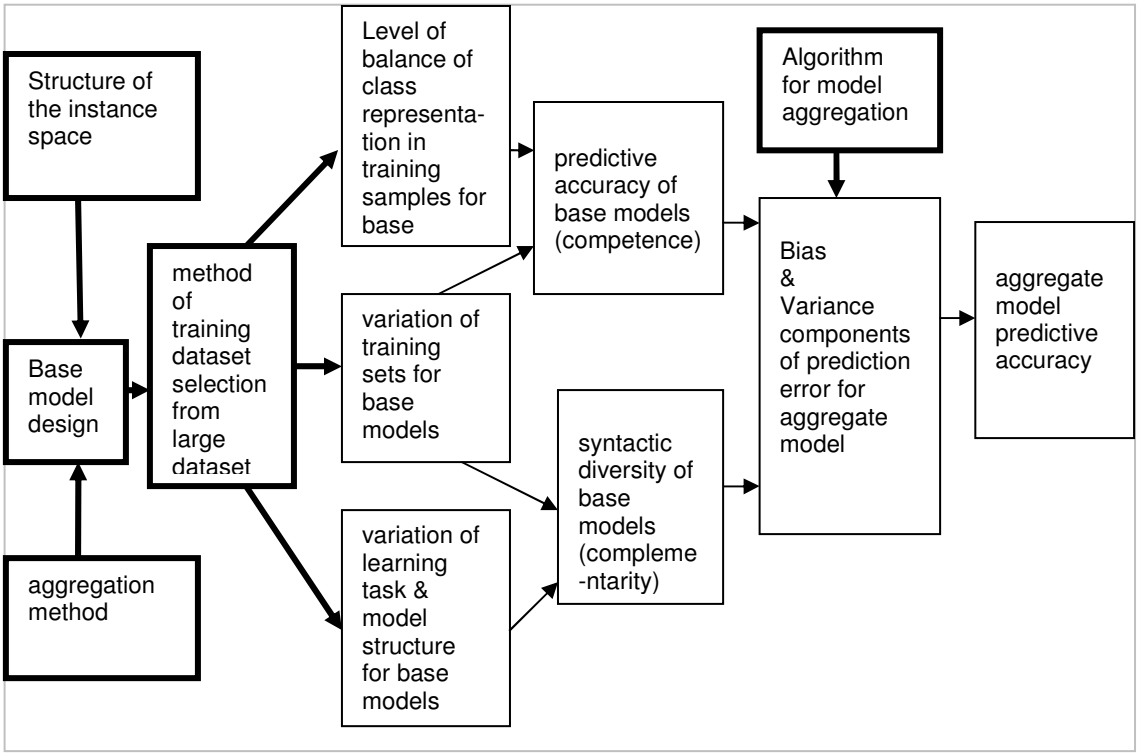

Figure 10.4: Extensions to the theoretical predictive model for aggregate model performance based on studies for this thesis

10.3.3 Parallel versus serial aggregation of base models

The method of parallel combination of base models was used to create the aggregate models for the experiments reported in this thesis. Serial combination (Sun & Li, 2008; Neagu et al, 2006; Kim et al, 2002) is the second method of base model aggregation which was discussed in chapter 2. Recall that s*erial combination* is a multi-step process. In the first step the base models are arranged in a series. In order to classify a new instance, the instance is passed to the first base model in the series. If the base model makes a *credible prediction,* then the process stops, otherwise the instance is passed to the next base model in the series. In general, if a base model makes a *credible prediction* the process stops otherwise the instance is passed to the next base model in the series (Sun & Li, 2008). The base model which provides the highest predictive accuracy on a given class is considered to be the base model that makes a *credible prediction* for that class (Sun & Li, 2008).

Sun and Li (2008) have conducted studies on the serial combination of base models, where each base model can make predictions for any of the classes and is constructed with a different classification algorithm. Two useful aspects of the base models were noted in the design and testing process for the pVn base models. First

of all, in general, a pVn base model was found to provide a higher level of accuracy on the positive classes that it predicts, compared to the single *k*-class model. Secondly, several of the classes for the prediction task can be predicted by more than one base model. Based on the foregoing observations the author hypothesized that the use of pVn base models in a serial combination scheme would provide performance improvements, especially for decision tree algorithms where no measures are available for resolving tied predictions. Studies to confirm this hypothesis were left for future work.

10.3.4 Guidelines for OVA and pVn model design, training dataset selection and testing

The steps given in figures 10.5 to 10.8 are recommended for the design of OVA and pVn aggregate models, training dataset design and selection for the models, and aggregate model creation and testing.

Phase I: Steps to establish class confusion

1. Partition the large dataset according to class, so that *k* partitions are created, one for each class.
2. From each partition obtained in step 1, set aside the data for model testing.
3. Decide on the sample size, *n*, for the creation of a single *k*-class model.
4. To obtain the training dataset for the single class model, proceed as follows. For each class c_i in the data, obtain a random sample of size n/k from the corresponding partition. If the partition has a size less than n/k, use bootstrap sampling to obtain the required sample size.
5. Combine all the samples obtained in step 4 to create the training dataset for the single *k*-class model. This training dataset will have an equal class distribution.
6. Create several test sets with an equal class distribution from the test data partitions.
7. Create the single *k*-class model and test it with the test sets created in step 6 in order to generate a confusion matrix for the classes.
8. Compute the predictive accuracy, TPRATE and TNRATE for each class in the single *k*-class model on the test sets.
9. If the confusion matrix is sparse, create a confusion graph from the confusion matrix.

Figure 10.5: Steps for the creation of a confusion matrix and confusion graph

The steps are based on the observations from the experiments of chapters 7 and 8. The first phase involves the establishment of the class confusion in a single *k*-class

model. Figure 10.5 shows the recommended steps to be followed for the identification of the class confusion.

Phase IIa: Un-boosted OVA model design, training dataset selection and testing

To create an un-boosted OVA aggregate model, proceed as follows:

1. Design the class and training sample composition for each OVA_i model so that class c_i has 50% of the instances and all the other classes combined have 50% of the instances.
2. Obtain the training samples for the OVA base models based on the design of step 1 by sampling from the partitions created in phase I. Use bootstrap sampling if the partition size is smaller than the required number of instances.
3. Create the OVA base models and OVA aggregate model, and test the aggregate model using the test samples created in phase I.
4. Compare the performance of the un-boosted OVA aggregate model with that of the single *k*-class model on the test samples.

Figure 10.6: Steps for the design, creation and testing of un-boosted OVA aggregate models

Phase IIb: Boosted OVA model design, training dataset selection and testing

If the single *k*-class model has a sparse confusion matrix, proceed as follows to create a boosted OVA aggregate model:

1. For each class c_i, determine from the confusion matrix or confusion graph which other classes are predominantly confused with c_i.
2. Design the class and training sample composition for each OVA_i model so that class c_i has 50% of the instances and the classes identified in step 1 have 50% of the instances. Consider apportioning the class representation based on the level of confusion, as discussed in chapter 7.
3. Obtain the training samples for the OVA base models based on the design of step 2 by sampling from the partitions created in phase I. Use bootstrap sampling it the partition size is smaller than the required number of instances.
4. Create the OVA base models and OVA aggregate model, and test the aggregate model using the test samples created in phase I.
5. Compare the performance of the OVA aggregate model with that of the single k-class model on the test samples.

Figure 10.7: Steps for the design, creation and testing of boosted OVA aggregate models

The steps given in figures 10.6 and 10.7 are recommended for purposes of creating OVA aggregate models. These steps should be conducted after the steps given in figure 10.5. For purposes of creating pVn aggregate models, the steps given in figure 10.8 are recommended. These steps should be conducted after the steps given in figure 10.5. Suggestions on how to conduct the steps of figures 10.6, 10.7 and 10.8 using commonly available software (SPSS and MS Excel) are given in the appendix table H.2.

Phase III: pVn model design, training dataset selection, and testing

If the single *k*-class model has a sparse confusion matrix, proceed as follows to create a pVn model:

1. Use the algorithms of figures 8.4 and 8.5 to establish the p-classes and n-classes for the base models, based on the confusion graphs created in phase I.
2. Design the training samples so that the p-classes combined have a high instance representation (eg. 80%) and the n-classes combined have a low instance representation (eg. 20%).
3. Obtain the training datasets designed in step 2 through random sampling from the partitions created in phase I.
4. Create the pVn base models and aggregate model and test the performance of the aggregate model using the test sets created in phase I.

Figure 10.8: Steps for the design, creation and testing of pVn aggregate models

10.5 Chapter summary

A summary of the research contributions for feature selection, base model design and training dataset selection have been given in this chapter. Recommendations for feature selection and training dataset selection for OVA and pVn modeling from large datasets have also been presented. A detailed discussion of the research contributions in terms of the expectations for design science research is provided in the next chapter.