Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 1220–1226
Expert Systems with Applications
www.elsevier.com/locate/eswa

Input data for decision trees

Selwyn Piramuthu *

Decision and Information Sciences University of Florida, Gainesville, FL 32611–7169, United States

Abstract

Data Mining has been successful in a wide variety of application areas for varied purposes. Data Mining itself is done using several different methods. Decision Trees are one of the popular methods that have been used for Data Mining purposes. Since the process of constructing these decision trees assume no distributional patterns in the data (non-parametric), characteristics of the input data are usually not given much attention. We consider some characteristics of input data and their effect on the learning performance of decision trees. Preliminary results indicate that the performance of decision trees can be improved with minor modifications of input data.

Keywords: Decision trees; Data characteristics

1. Introduction

Data Mining has been successful in a wide variety of application areas, including marketing, for varied purposes (Adomavicius & Tuzhilin, 2001; Kushmerick, 1999; van der Putten, 1999; Shaw, Subramaniam, Tan, & Welge, 2001; Thearling, 1999). Data Mining itself is done using several different methods, depending on the type of data as well as the purpose of Data Mining (Ansari, Kohavi, Mason, & Zheng, 2000; Cooley, Tan, & Srivastava, 1999; Srivastava, Cooley, Deshpande, & Tan, 2000). For example, if the purpose is classification using real data, feed-forward neural networks might be appropriate (Ragavan & Piramuthu, 1991). Decision trees might be appropriate if the purpose is classification using nominal data (Quinlan, 1993). Further, if the purpose is to identify associations in data, association rules might be appropriate (Brijs, Swinnen, Vanhoof, & Wets, 1999).

Decision Trees are one of the popular methods that have been used for Data Mining purposes. Decision trees can be constructed using a variety of methods. For example, C4.5 (Quinlan, 1993) uses information-theoretic measures and CART (Breiman, Friedman, Olshen, & Stone, 1984) uses statistical methods. The usefulness as well as classification and computational performance of Data Mining frameworks incorporating decision trees can be improved by (1) appropriate preprocessing of input data, (2) fine-tuning the decision tree algorithm itself, and (3) better interpretation of output. There have been several studies that have addressed each of these scenarios.

Input data can be preprocessed (1) to reduce the complexity of data for ease of learning, and (2) to reduce effects due to unwanted characteristics of data. The former includes such techniques as feature selection and feature construction as well as other data modifications (see, for example, Brijs & Vanhoof, 1998; Kohavi, 1995; Ragavan & Piramuthu, 1991). The latter includes removal of noisy, redundant, and irrelevant data used as input to decision tree learning.

We consider some characteristics of input data and its effect on the learning performance of decision trees. Specifically, we consider the effects of non-linearity, outliers, heteroschedasticity, and multicollinearity in data. These have been shown to have significant effects on regression analysis. However, there has not been any published study that deals with these characteristics and their effects on the learning performance of decision trees. Using a few small data sets that are available over the Internet, we consider each of these characteristics and compare their effects on regression

* Tel.: +1 352 392 8882; fax: +1 352 392 5438.
E-mail address: selwyn@ufl.edu

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.12.030

analysis as well as decision trees. The results from regression analysis are from the Internet. The contribution of this paper is in studying the effects of these characteristics on decision trees, specifically See-5 (2001). Preliminary results suggest that the performance of decision trees can be improved with minor modifications of input data.

The rest of the paper is organized as follows: Evaluation of some input data characteristics and their effects on the learning performance of decision trees is provided in the next section. Experimental results are also included in Section 2. Section 3 concludes the paper with a brief discussion of the results from this study and their implications as well as future extensions to this study.

2. Evaluation of input data characteristics for decision trees

Traditional statistical regression analysis assumes certain distribution (e.g., Gaussian) of input data, as well as other characteristics of data such as the data being independent and identically distributed. In most real world data, some of these assumptions are often violated. And, there are several means to at least partially rectify some of the consequences that arise from these violations. We consider a few of these situations: non-linearity in the data, the presence of outliers in the data, the presence of heteroschedasticity in the data, and the presence of multicollinearity in the data. The data sets used in this study are known to have these characteristics. The following subsections address each of these scenarios in turn. We use See-5 as the decision tree generator throughout this study.

2.1. Non-linearity in input data

Non-linearity is a problem in linear regression simply because it is hard to fit a linear model on a non-linear data. Therefore, non-linear transformations are made to the data before running regressions on these data. We consider the effects of non-linear data on decision trees both before and after the appropriate data transformations are made.

This data set contains four variables – the independent variables x_1, x_2, and x_3 and the dependent variable y. Result using ordinary least squares (OLS) regression to predict y using x_1, x_2, and x_3 are provided below.

Source	SS	df	MS	Number of obs = 100		
				$F(3,96) = 2.21$		
Model	5649.25003	3	1883.08334	Prob > $F = 0.0915$		
Residual	81668.75	96	850.716146	$R^2 = 0.0647$		
				Adj $R^2 = 0.0355$		
Total	87318.00	99	882.00	Root MSE = 29.167		
y	Coef.	Std. Err.	t	$P > \|t\|$	(95% Conf. interval)	
x_1	.1134093	.3626687	0.313	0.755	−.06064824	.833301
x_2	−.0089643	.3757505	−0.024	0.981	−.7548232	.7368946
x_3	.5932696	.2560351	2.317	0.023	.0850430	1.101495
-cons	20.09967	11.61974	1.730	0.087	−2.965335	43.16468

The presence of higher order trend effects are identified to be present in the data using the omitted variable test (ovtest with the rhs option in the statistical analysis software *Stata*). Higher order trend effects are also present. On inspection of scatter plots of the data, the presence of non-linear trend patterns in the data in the variable x_2 is confirmed. We substitute x_2 with its centered ($x_2$cent) value (i.e., subtract its mean from every value) and the square ($x_2$centsq) of the centered value. The results from this regression are provided below:

Source	SS	df	MS	Number of obs = 100		
				$F(4,95) = 171.00$		
Model	76669.37633	4	19167.3441	Prob > $F = 0.0000$		
Residual	10648.6237	95	112.090775	$R^2 = 0.08780$		
				Adj $R^2 = 0.8729$		
Total	87318.00	99	88.2	Root MSE = 10.587		
y	Coef.	Std. Err.	t	$P > \|t\|$	(95% Conf. interval)	
x_1	.1706278	.1316641	1.296	0.198	−.0907856	.4320141
$x_2$cent	−.0262898	.1363948	−0.193	0.848	−.2970677	.244488
$x_2$centsq	.2954615	.11738	25.171	0.000	.2721586	.3187645
x_3	.2584843	.0938846	2.753	0.007	.0720997	.4448688
-cons	−.8589132	1.3437	−0.639	0.524	−3.526496	1.808669

On further testing for higher order terms, the result turns out to be negative. Here, by including the squared term, it is shown that the new term is indeed statistically significant. The resulting model also fits the data better as shown by the increase in the R^2 value.

Now, let us consider the same two sets of data, both before and after incorporating the squared term, and evaluate its effects on the performance of decision tree learned. We use 10-fold cross-validation in See-5 to reduce any bias due to sample selection. Both the mean values and the standard deviation values (in parentheses) are provided for the resulting decision trees.

Table 1
Results using non-linear data

Input variables	Decision tree size	Prediction error (%)
x_1, x_2, x_3	3.2 (0.6)	28.0 (4.2)
x_1, $x_2$cent, $x_2$centsq, x_3	3.0 (0.5)	11 (2.8)

This data set used here contains four variables – the independent variables x_1, x_2, and x_3 and the dependent variable y. Result using OLS regression to predict y using x_1, x_2, and x_3 are provided below.

Source	SS	df	MS	Number of obs = 100		
				$F(3,96) = 14.12$		
Model	6358.64512	3	2119.54837	Prob > $F = 0.0000$		
Residual	14406.3149	96	150.06578	$R^2 = 0.3062$		
				Adj $R^2 = 0.2845$		
Total	20764.96	99	209.747071	Root MSE = 12.25		
y	Coef.	Std. Err.	t	$P > \|t\|$	(95% Conf. interval)	
x_1	.1986327	.1523206	1.304	0.195	−.1037212	.5009867
x_2	.576853	.1578149	3.655	0.000	.2635928	.8901132
x_3	.3533915	.1075346	3.286	0.001	.1399371	.5668459
-cons	32.33932	1.229643	26.300	0.000	29.8985	34.78014

Here (Table 1), the addition of the two transformed x_2 variables has resulted in a small reduction in the size of the decision trees and a significant decrease in the prediction error. The prediction error is the classification error on unseen (during generation of decision trees) examples.

2.2. Presence of outliers in input data

The presence of outliers in input data is a problem in any learning application because most methods used for learning patterns in the data over-fit the outliers. Depending on how serious the outliers are the resulting patterns learned could be significantly different from the actual patterns without the outliers. We consider the effects of outliers on decision trees both before and after the outliers are removed.

We start out to examine for the presence of outliers by looking at the scatter plot of the data. On inspection, we identify a single point that stands out from the rest. After several attempts at evaluating this point, we find that this point indeed is an outlier. This data point is due to a typographical error during data input, and therefore the error that led to this situation is corrected. The result from the OLS regression run on the data with the outlier problem fixed is provided below.

Source	SS	df	MS	Number of obs = 100		
				$F(3,96) = 20.27$		
Model	5863.70256	3	1954.56752	Prob > $F = 0.0000$		
Residual	9255.28744	96	96.4092442	$R^2 = 0.3878$		
				Adj $R^2 = 0.3687$		
Total	15118.99	99	152.717071	Root MSE = 9.8188		
y	Coef.	Std. Err.	t	$P > \|t\|$	(95% Conf. interval)	
x_1	.325315	.1220892	2.665	0.009	.08297	.5676601
x_2	.4193103	.126493	3.315	0.001	.1682236	.670397
x_3	.3448104	0.861919	4.000	0.000	.1737208	.5159
-cons	31.28053	.9855929	31.738	0.000	29.32415	33.23692

Now, let us consider the same two sets of data, both before and after fixing the problem of outlier data, and evaluate its effects on the performance of decision tree learned. Again, we use 10-fold cross-validation in See-5 to reduce any bias due to sample selection.

Table 2
Results using outlier data

Input variables	Decision tree size	Prediction error (%)
x_1, x_2, x_3 (with outlier)	4.8 (0.6)	46.0 (5.0)
x_1, x_2, x_3 (without outlier)	4.4 (0.6)	38 (4.2)

Here (Table 2), the removal of the input error has resulted in a small reduction in the size of the decision trees and a significant decrease in the prediction error.

2.3. Heteroschedasticity in input data

Data where the error term variance is not constant has Heteroschedasticity. The presence of heteroschedasticity in input data is a problem in regression analysis because the modeling depends on the assumption that that the error term variance is a constant. We consider the effects of heteroschedasticity on decision trees both before and after the problem has been alleviated in the input data.

This data set contains four variables – the independent variables x_1, x_2, and x_3 and the dependent variable y. Result using OLS regression to predict y using x_1, x_2, and x_3 are provided below.

Source	SS	df	MS	Number of obs = 100		
				$F(3,96) = 65.68$		
Model	8933.72373	3	2977.90791	Prob > $F = 0.0000$		
Residual	4352.46627	96	45.3381903	$R^2 = 0.6724$		
				Adj $R^2 = 0.6622$		
Total	13286.19	99	134.203939	Root MSE = 6.7334		
y	Coef.	Std. Err.	t	$P > \|t\|$	(95% Conf. interval)	
x_1	.2158539	.83724	2.578	0.011	.0496631	.3820447
x_2	.7559357	.086744	8.715	0.000	.5837503	.9281211
x_3	.3732164	.0591071	6.314	0.000	.2558898	.490543
-cons	33.23969	.6758811	49.180	0.000	31.89807	34.5813

We use the hettest command in *Stata* to test for heteroschedasticity, and find that the results are indeed heteroschedastic. We then try to stabilize the variance by using a natural logarithmic transformation of the dependent variable. The OLS regression results after this transformation is provided below:

Source	SS	df	MS	Number of obs = 100		
				$F(3,96) = 69.85$		
Model	8.17710164	3	2.72570055	Prob > $F = 0.0000$		
Residual	3.74606877	96	.03902155	$R^2 = 0.6858$		
				Adj $R^2 = 0.6760$		
Total	11.9231704	99	.120436065	Root MSE = .19754		
y	Coef.	Std. Err.	t	$P > \|t\|$	(95% Conf. interval)	
x_1	.0054677	.0024562	2.226	0.028	.0005921	.0103432
x_2	.0230303	.0025448	9.050	0.000	.0179788	.0280817
x_3	0.118223	.001734	68.18	0.000	.0083803	.0152643
-cons	3.445503	.0198285	173.765	0.000	3.406144	3.484862

Table 3
Results using heteroschedasticity data

Input variables	Decision tree size	Prediction error (%)
x_1, x_2, x_3 (dep var: y)	4.9 (0.6)	27.0 (3.0)
x_1, x_2, x_3 (dep. var: ln(y))	5.5 (0.4)	26 (4.0)

Now, let us consider the same two sets of data, both before and after fixing the problem of heteroschedasticity, and evaluate its effects on the performance of decision tree learned. Again, we use 10-fold cross-validation in See-5 to reduce any bias due to sample selection.

Here (Table 3), the transformation of the dependent variable has resulted in a small increase in the size of the decision trees and an insignificant decrease in the prediction error.

2.4. Multicollinearity in input data

Data where the independent variables are highly correlated is said to have multi-collinearity. In regression analysis, multicollinearity is a problem when we are interested in the exact values of the coefficients of the independent variables. When multicollinearity is present, this is not possible. Multicollinearity is identified by (1) the presence of high pairwise correlation among independent variables, (2) a high R^2 value with low t-statistics, and (3) the coefficients change when variables are added and dropped from the model. Multicollinearity is not a problem when the only purpose of regression analysis is forecasting. However, if the analysis is to determine and evaluate the coefficients, multicollinearity

is a problem. One of the ways to alleviate this problem is to drop the variable with the highest pair-wise correlation values among the independent variables. We consider the effects of multicollinearity on decision trees both before and after the problem has been alleviated in the input data.

This data set contains five variables – the independent variables x_1, x_2, x_3, and x_4 and the dependent variable y. Result using OLS regression to predict y using x_1, x_2, x_3, and x_4 are provided below.

Source	SS	df	MS	Number of obs = 100		
				$F(3,96) = 16.37$		
Model	5995.66253	4	1498.91563	Prob > $F = 0.0000$		
Residual	8699.33747	95	91.5719733	$R^2 = 0.4080$		
				Adj $R^2 = 0.3831$		
Total	14695.00	99	148.434343	Root MSE = 9.5693		
y	Coef.	Std. Err.	t	$P > \|t\|$	(95% Conf. interval)	
x_1	1.118277	1.024484	1.092	0.278	−.9155806	3.152135
x_2	1.286694	1.042406	1.234	0.220	−.7827429	3.356131
x_3	1.191635	1.05215	1.133	0.260	−.8971469	3.280417
x_4	−.8370988	1.038979	−0.806	0.422	−2.899733	1.225535
-cons	31.61912	.9709127	32.566	0.000	29.69161	33.54662

Here, the R^2 value is significant while the t-statistics are not. We then consider the pair-wise correlations among the independent variables. The resulting matrix is given below:

	x_1	x_2	x_3	x_4
x_1	1.0000			
x_2	0.3553	1.0000		
x_3	0.3136	0.2021	1.0000	
x_4	0.7281	0.6516	0.7790	1.0000

Clearly, x_4 is highly correlated with the rest of the independent variables. This variable is then removed from the data. The results from the OLS regression run without x_4 is given below:

Source	SS	df	MS	Number of obs = 100		
				$F(3,96) = 21.69$		
Model	5936.21931	3	1978.73977	Prob > $F = 0.0000$		
Residual	8758.78069	96	91.2372989	$R^2 = 0.4040$		
				Adj $R^2 = 0.3853$		
Total	14695.00	99	148.434343	Root MSE=9.5518		
y	Coef.	Std. Err.	t	$P > \|t\|$	(95% Conf. interval)	
x_1	.298443	.1187692	2.513	0.014	.0626879	.5341981
x_2	.4527284	.1230534	3.679	0.000	.2084695	.6969874
x_3	.3466306	.0838481	4.134	0.000	.1801934	.5130679
-cons	31.50512	.9587921	32.859	0.000	29.60194	33.40831

Here, all the variables turn out to be significant.

Now, let us consider the same two sets of data, both before and after removing x_4 from the data, and evaluate its effects on the performance of decision tree learned. Again, we use 10-fold cross-validation in See-5 to reduce any bias due to sample selection.

Table 4
Results using multicollinearity data

Input variables	Decision tree size	Prediction error (%)
x_1, x_2, x_3, x_4	3.2 (0.5)	34.0 (3.1)
x_1, x_2, x_3	4.8 (0.5)	45.0 (2.7)

Here (Table 4), the transformation of the dependent variable has resulted in a significant increase in the size of the decision trees and a significant increase in the prediction error. Here, removal of the variable does not seem to help the performance of decision trees.

2.5. Data reduction

Data reduction is an important preprocessing step in any pattern recognition method. Benefits of data reduction include removal of irrelevant attributes from data, alleviate effects due to outliers in data, reduce effects due to noise, resulting parsimony in decision-making, reducing data complexity for learning algorithms, among others. These are even more critical when dealing with huge data sets, as is the case with most data mining applications. Data reduction essentially involves dimensionality reduction and/or example reduction, among others. In a majority of

cases, when dealing with data used as input to learning algorithms, the former deals with reducing the number of relevant attributes and the latter involves effectively reducing the number of examples. In this study, we present and evaluate a framework to reduce the effective number of examples that are used as input to pattern learning algorithms.

Most example reduction methods use some form of sampling (e.g., random, stratified) to select examples to be considered further. The underlying assumptions are different for each of these sampling methods. For example, in simple random sampling, it is assumed that every unit in the population provides the same amount of information. There is a vast amount of literature on example reduction methods (e.g. Ishibuchi, Nakashima, & Nil, 2001; Liu & Motoda, 2001; Provost & Kolluri, 1999; Provost, Jensen, & Oates, 1999).

We utilize clustering as a pre-processing step for learning applications. Specifically, we use (k-means) clustering for data reduction in the number of example dimension, and as a pre-processing step for decision trees. We use the fuzzy thresholds option in See-5 since we are using real-numbered variables in the data set. See-5 partitions real-numbered variable at a given threshold point, and small movements in the variable value near the threshold can change the branch taken. The Fuzzy thresholds option softens this knife-edge behavior for decision trees by constructing an interval close to the threshold. Within this interval, both branches of the tree are explored and the results combined to give a predicted class.

We use the Iris plants database (Anderson, 1935; Fisher, 1936) to illustrate the proposed method. We chose this database simply because it is among the best known pattern recognition databases (e.g. Duda & Hart, 1973), its simplicity, and any results generated using this database can be readily compared with those generated through other methods. The database consists of 150 examples covering three iris plant types (Iris-Setosa, Iris-Versicolor, and Iris-Virginica), with 50 examples from each type. Data from one of the type is linearly separable from the other two, and the latter two are not linearly separable from each other. The four independent attributes (sepal length, sepal width, petal length, and petal width) are numeric. There are no missing attribute values.

Since we want to compare the results for the case where clustering is used with that where clustering was not used as a pre-processing step, we randomly divided the data set into five parts (a,b,c,d,e). We did this to alleviate problems due to sampling errors. The five data sets were then used in a leave-one-out fashion. I.e., in the first case, the first four parts were used to generate the clusters (abcd), which were then used as input to See-5. The resulting decision tree was tested using the last part (e). We repeated this five times for each case, where different parts were used to test the resulting decision tree.

Table 5 summarizes the results based on the number of examples used as input to the decision tree generator (See-5). For training error, testing error and tree size, the mean value is followed by standard deviation values in parentheses. As can be seen neither the training error nor the testing error seem to be uni-modal functions. However, there is a significant decrease in both training (pair-wise two-tailed t-test with $p < 0.005$) and testing (pair-wise two-tailed t-test with $p = 0.026$) error as the number of clusters decreases. This could possibly be because the benefits of clustering become apparent as the number of examples used to form any given cluster is increased. In the example data set used, the best results are for data sets with 15 and 9 clusters. Then again, as the number of clusters is reduced to its minimum (here, two for each iris type, resulting in 6 clusters overall), there tends to be a loss in information content simply because of the complexity of data dictating the necessity for more points (here, clusters) to draw boundaries among examples belonging to different categories (here, type of iris). The decision tree size also decreases (statistically significant with a pair-wise two-tailed t-test with $p < 0.002$) as the number of clusters is decreased.

Table 5
Summary results for data reduction

# of training examples	Training error	Testing error	Tree size
120	2 (0.94)	4.66 (3.81)	4.6(0.55)
60	3.68 (2.17)	0 (0)	3.4 (0.55)
30	3.32 (2.37)	4 (2.81)	3 (0)
15	0 (0)	0 (0)	3 (0)
9	0 (0)	0 (0)	3 (0)
6	0 (0)	5.34 (7.69)	3 (0)

3. Discussion

Even though decision trees constructed using information-theoretic measures are considered non-parametric, the distribution of data does influence the classification performance of these decision trees. Preliminary results indicate that the performance of decision trees can be improved by considering the effects due to non-linearity, outliers, heteroschedasticity, and multicollinearity in input data as well as data reduction. Both non-linearity and the presence of outliers did affect the classification performance of decision trees. The presence of heteroschedasticity did not affect the classification performance of decision trees significantly. And, the presence of multicollinearity is not of concern for decision trees. The attempt to remove multicollinearity resulted in poor classification performance. Data reduction resulted in improved performance both in terms of the resulting tree-size and classification.

We are currently in the process of evaluating the results we have thus far using larger and more data sets. We are also in the process of studying why these data characteristics affect the classification performance of decision trees. Also, in this study, we were only interested in the size of the decision trees and their classification accuracy. The computational cost of this process is also important, and this is left as an exercise for a future study. In addition to the characteristics presented in this paper, we are

also evaluating other data characteristics including non-independence and non-normality of data.

We presented one possible means to improve the classification performance of decision trees. This, along with other pre-processing methods (such as feature selection and feature construction), methods for fine-tuning decision trees, and those that enhance interpretability of results, would help improve the overall performance of these decision support tools incorporating decision trees.

References

Adomavicius, G., & Tuzhilin, A. (2001). Using data mining methods to build customer profiles. *IEEE Computer* (February), 74–82.

Anderson, E. (1935). The Irises of the Gaspe Peninsula. *Bulletin of the American Iris Society, 59*, 2–5.

Ansari, S., Kohavi, R., Mason, L., & Zheng, Z. (2000). Integrating E-commerce and data mining: architecture and challenges. *WEB-KDD'2000 workshop on Web mining for E-commerce – challenges and opportunities*, August.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression*. Wadsworth.

Brijs, T., & Vanhoof, K. (1998). Principles of data mining and knowledge discovery. In J. M. Zytkow (Ed.), *PKDD '98. Lecture notes in artificial intelligence* (Vol. 1510, pp. 102–110). Berlin: Springer.

Brijs, T., Swinnen, G., Vanhoof, K., & Wets, G. (1999). Using association rules for product assortment decisions: a case study. In *Proceedings of the 5th international conference on knowledge discovery and data mining (KDD'99)* (pp. 254–260). San Diego, CA, August 15–18, 1999.

Cooley, R., Tan, P.-N. & Srivastava, J. (1999). Discovery of interesting usage patterns from Web data. *Technical Report*, Department of Computer Science and Engineering, University of Minnesota.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. John Wiley & Sons (p. 218).

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annual Eugenics, 7*(Part II), 179–188.

Ishibuchi, H., Nakashima, T., & Nil, M. (2001). Genetic-algorithm-based instance and feature selection. In H. Liu & H. Motoda (Eds.), *Instance selection and construction for data mining*. Kluwer Academic.

Kohavi, R. (1995). *Wrappers for performance enhancement and oblivious decision graphs*, Ph.D. Dissertation, Computer Science Department, Stanford University.

Kushmerick, N. (1999). Learning to remove Internet advertisements. *Third International Conference on Autonomous Agents*.

Liu, H., & Motoda, H. (Eds.). (2001). *Instance selection and construction for data mining*. Kluwer Academic.

Provost, F., Jensen, D., & Oates, T. (1999). Efficient progressive sampling. In *Proceedings of the 5th international conference on knowledge discovery and data mining* (pp. 23–32). AAAI Press.

Provost, F., & Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery, 3*(2), 131–169.

Quinlan, J. R. (1993). *C4.5 programs for machine learning*. Morgan Kaufman.

Ragavan, H., & Piramuthu, S. (1991). The Utility of Feature Construction in Back-propagation. *Proceedings of the twelfth IJCAI*, 844–848.

See-5. (2001). Rulequest research data mining tools.

Shaw, M., Subramaniam, C., Tan, G. W., & Welge, M. E. (2001). Knowledge management and data mining for marketing. *Decision Support Systems, 31*, 127–137.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P.-N. (2000). Web usage mining: discovery and applications of usage patterns from Web data. *SIGKDD Explorations, 1*(2), 12–23, January.

Thearling, K. (1999). Data mining and CRM: zeroing in on your best customers. *dmDitrect* (December), 20.

van der Putten, P. (1999). Data mining in direct marketing databases. In W. Baets (Ed.), *Complexity and management: a collection of essays*. Singapore: World Scientific Publishers.