Performance Controlled Data Reduction for Knowledge Discovery in Distributed Databases

Slobodan Vucetic and Zoran Obradovic

School of Electrical Engineering and Computer Science
Washington State University, Pullman, WA 99164-2752, USA
{svucetic, zoran}@eecs.wsu.edu

Abstract. The objective of data reduction is to obtain a compact representation of a large data set to facilitate repeated use of non-redundant information with complex and slow learning algorithms and to allow efficient data transfer and storage. For a user-controllable allowed accuracy loss we propose an effective data reduction procedure based on guided sampling for identifying a minimal size representative subset, followed by a model-sensitivity analysis for determining an appropriate compression level for each attribute. Experiments were performed on 3 large data sets and, depending on an allowed accuracy loss margin ranging from 1% to 5% of the ideal generalization, the achieved compression rates ranged between 95 and 12,500 times. These results indicate that transferring reduced data sets from multiple locations to a centralized site for an efficient and accurate knowledge discovery might often be possible in practice.

Keywords: data reduction, data compression, sensitivity analysis, distributed databases, neural networks, learning curve

1 Introduction

An important knowledge discovery problem is to establish a reasonable upper bound on the size of a data set needed for an accurate and efficient analysis. For example, for many applications increasing the data set size 10 times for a possible accuracy gain of 1% can not justify huge additional computational costs. Also, overly large training data sets can result in increasingly complex models that do not generalize well [8].

Reducing large data sets into more compact representative subsets while retaining essentially the same extractable knowledge could speed up learning and reduce storage requirements. In addition, it could allow application of more powerful but slower modeling algorithms (e.g. neural networks) as attractive alternatives for discovering more interesting knowledge from data.

Data reduction can be extremely helpful for data mining on large distributed data sets where one of the more successful current approaches is learning local models at each data site, and combining them in a meta-model [11]. The advantage of meta-modeling is that learning local models and integrating them is computationally much more efficient than moving large amounts of data into a centralized memory for learning global models. However, this sub-optimal heuristic assumes similarity between local data sets and it is not clear how to successfully combine local models learned on data with different distributions and not identical sets of attributes.

A centralized approach of transferring all data to a common location escapes the sub-optimality problems of local models combination, but is often infeasible in practice due to a limited communication bandwidth among sites. Reducing data sets by several orders of magnitude and without much loss of extractable information could speed up the data transfer for a more efficient and a more accurate centralized learning and knowledge extraction. Here, for user-specified allowed accuracy loss we propose an effective data reduction procedure based on applying a guided sampling to identify a minimal size representative sample followed by a model-sensitivity analysis to determine an appropriate compression level for each attribute.

The problem addressed in this study is more formally defined in Section 2, the proposed data reduction procedure is described in Sections 3-4 and an application to 3 large data sets is reported in Section 5.

2 Definitions and Problem Description

To properly describe the goal of data reduction and to explain the proposed approach, some definitions will be given first. The definitions apply to regression and classification problems solved by learning algorithms minimizing least square error, including linear models and feedforward neural networks.

Definitions

Given a data set with N examples, each represented by a set of K attributes, $\mathbf{x}=\{x_1,\ldots,x_K\}$, and the corresponding target y, we denote the underlying relationship as $y = E[y|\mathbf{x}]+\varepsilon$, where ε is an additive error term. For regression problems the target is usually a single number, while for L-class classification problems it is usually an L-dimensional vector. We define the *reduced data set* as any data set obtained from the given one by (1) reduction of the number of examples called *down-sampling*, or/and (2) quantization of its attributes and targets. The *length* of a data set is defined as the number of bits needed for its representation. *Compression* rate *C* equals the ratio between the bitwise length of the original data set and the bitwise length of the reduced data set.

Assuming a parametric learning algorithm, by $f(\mathbf{x}; \beta(n))$ we denote a predictor learned on n examples and we measure its performance by the mean squared error (MSE) defined as $\text{MSE}(\beta)=E\{[y- f(\mathbf{x}; \beta)]^2\}$, where β is the set of the model parameters. If a predictor is learned on a reduced, instead of the original, data set some increase in the MSE is to be expected. The total *relative MSE increase*, $\text{MSE}(\beta_q(n))/\text{MSE}(\beta(N))$, where $\beta_q(n)$ are estimators of parameters from a model learned on a down-sampled data set with quantized attributes, is the product of relative MSE increases due to down-sampling $\text{MSE}(\beta(n))/\text{MSE}(\beta(N))$ and quantization $\text{MSE}(\beta_q(n))/\text{MSE}(\beta(n))$. Throughout the text we denote the relative MSE increase as $(1+\alpha)$, and call α the *loss margin*.

Problem Description

Our goal is to obtain a minimal length reduced data set that, using the same learning algorithm, allows extraction of the same knowledge reduced for at most a loss margin α. To achieve this goal we propose two successive phases: (1) reduce the sample size from N to n_{min} allowing loss margin α_D and achieving compression $C_D=N/n_{min}$, and (2) perform proper quantization of attributes of a down-sampled data set allowing loss margin α_Q, followed by Huffman coding [5] of discretized attributes and achieving compression C_Q. Assuming that total loss margin α is close to zero, it follows that $\alpha \approx \alpha_D+\alpha_C$ with an achieved total compression $C = C_DC_Q$. To keep the presentation simple, we will assume that $\alpha_D = \alpha_C = \alpha/2$, and will skip the optimization of α_D and α_C for the maximum achievable total compression.

The motivation for data reduction is obtaining the compact representation of a data set that would facilitate its efficient repeated use with complex learning algorithms and allow its efficient transfer and storage. All these features are highly desirable for data mining on distributed databases. In this framework, local computing time needed for data reduction would not be a critical requirement, since this effort would be rewarded multifold. Nevertheless, for large data sets, the whole data reduction effort has comparable or even lower computational time as compared to building a single model on a whole data set. In the following two sections we separately describe procedures for down-sampling and quantization and compression.

3 Identifying a Minimum Representative Sample

3.1 Down-Sampling for Fast and Stable Algorithms

The *learning curve* for least squares estimators shows the average MSE dependence on the size n of a sample used for designing estimators. This curve can be divided into an *initial region* characterized by the fast drop of the MSE with increasing sample size and a *convergence region* where addition of new samples is not likely to significantly improve prediction (see Fig. 1). The learning curve is the result of complex relationships between data characteristics and a learning algorithm. Therefore its shape needs to be determined by experimentation with an objective of identifying size n_{min} of a minimum representative sample needed to achieve an approximation of the optimal average MSE within a specific *loss margin* α_D.

An asymptotic analysis based on the law of large numbers and the central limit theorem [4] can help in successful modeling of a learning curve. According to the asymptotic normality theorem for nonlinear least squares, estimation error is asymptotically normal under certain fairly general conditions. Asymptotic normality means that $n^{1/2}(\beta(n)-\beta^*)$ tends in distribution to a normal distribution with mean zero and finite covariance matrix, where $\beta(n)$ is an n-sample based estimate of the true parameter vector β^*. The consequence is that for large n, residuals ε' of the nonlinear estimator $f(x;\beta(n))$ consistently estimate the actual disturbances as $\varepsilon' = \varepsilon + O(n^{-1/2})$. Therefore,

$$\mathrm{MSE}(\beta(n)) = \mathrm{MSE}(\beta^*) + u, \quad u \sim N(O(1/n), O(1/n)), \tag{1}$$

meaning that MSE($\beta(n)$) asymptotically tends to the optimum MSE(β^*) as $O(1/n)$, with variance decreasing as $O(1/n)$. Assuming that N corresponds to the convergence region and using (1), modeling of a learning curve to estimate a minimum representative sample size n_{min} can be fairly straightforward.

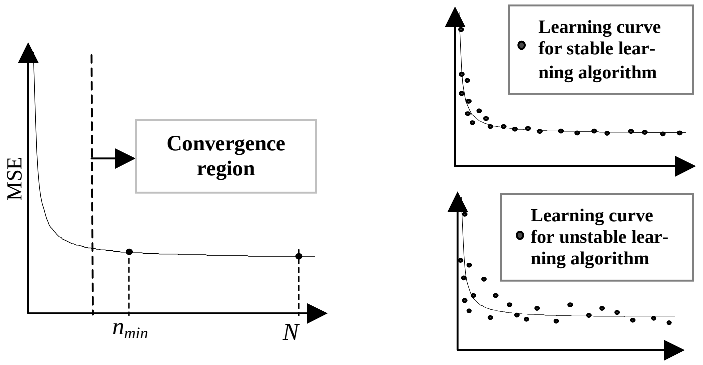

Fig. 1. Learning Curve

A recently proposed progressive sampling procedure [9] can efficiently span the available range of sampling sizes in search for the n_{min}. The technique was developed with an objective of increasing the speed of inductive learning by providing approximately the same accuracy and using significantly smaller sample sizes than available. It was shown that geometrical progressive sampling that starts with a small sample and uses geometrically larger sample sizes until exceeding n_{min} (model accuracy no longer improves) is an asymptotically optimal sampling schedule. We use the idea of progressive sampling with a somewhat different motivation of guiding an efficient search for a minimal sample size needed for achieving an approximation of the optimal average MSE within a specific *loss margin* α_D.

In regression statistical theory it is well known that linear least squares algorithms are the optimal estimators for linear problem solving. They are characterized by fast learning with time complexity $O(n)$ and well-known statistical properties including small variance. The following *DS1* procedure is proposed for identifying n_{min} value for fast and stable models:

- Estimate model on the whole available data set of size N and calculate MSE(N);
- Estimate model on a small sample of size n_0 and calculate MSE(n_0);
- Increase the sampling size from n_i to $n_i \cdot a^i$ until a sample size n_{min} is reached satisfying MSE(n_{min}) < (1+α_D)MSE(N).

A direct consequence of the progressive sampling results [9] for models with time complexity $O(n)$ is that the time complexity of this procedure for a=2 is at most twice the time of learning on the whole data set. This procedure might also be used for simple nonlinear algorithms with small variance (e.g. the feedforward neural networks without hidden nodes).

3.2 Down-Sampling Extension for Slower and Unstable Algorithms

Complex nonlinear learning algorithms such as feedforward neural networks with a number of hidden nodes typically have a large variance meaning that their MSE($\beta(n)$) can largely differ over different weight's initial conditions and choice of training data. Using explained *DS1* down-sampling procedure for such learning algorithms could cause significant errors in the estimation of n_{min}. Also, with these algorithms learning time for large N can be so long that the cost of obtaining a benchmark value of MSE(N) is unacceptable.

Using (1) and assuming that N is within a learning curve convergence region down-sampling can be performed by fitting learning curve samples obtained through guided sampling as

$$MSE(n) = \gamma_0 + \gamma_1 / n + \gamma_2 / n^2 + e, \quad e \sim \mathrm{N}(0, O(1/n))\,, \tag{2}$$

where γ_0 corresponds to an estimate of MSE for an infinitely large dataset, γ_1 to O(1/n) part of (1), and γ_2 to the initial region of a learning curve.

The error variance of the learning curve samples decreases as O(1/n), and so larger confidence should be given to MSE's of estimators learned on larger data samples. Therefore, we apply a weighted least squares algorithm [6] to fit the learning curve by multiplying (2) by $n^{1/2}$ and learning γ's on transformed learning curve samples.

For slower and unstable algorithms we propose the following down-sampling procedure that we will call *DS2*:

- Starting from a sample of size n_0, at iteration i increase sample size to $n_i = n_0 \cdot a^i$ until t-statistics for γ_0 and γ_1 do not exceed $t_{\theta,n-3}$ for some tolerant confidence level θ;
- Repeat until the difference in estimating n_{min} over several successive iterations is sufficiently small:
    - According to estimated γ's and predetermined loss margin, α_D, estimate n_{min} using (2);
    - Select the next sample size n_{i+1} larger then n_{min}. Larger n_{i+1} results in a larger improvement in the estimation of n_{min}, but at increased computational cost. Our heuristic of randomly selecting n_{i+1} within an interval [n_{min}, 2n_{min}] has proven to be a good compromise;
    - Learn a new model on n_{i+1} samples and calculate its MSE($\beta(n_{i+1})$);
- Output the last estimated n_{min} as the minimum representative sample size.

If neural networks are used in the down-sampling procedure, the minimum sample size is selected larger than the estimated value since part of the data should be reserved to validation subset. Our heuristic determines the total representative size as 1.5n_{min} such that in all iterations of down-sampling algorithm, 0.5n_i samples are being used as a validation set for an early stopping of neural network training.

4 Compression of a Minimum Representative Sample

Storing continuous attributes usually assumes double precision (8 bytes) for an accurate representation. Performing quantization of a continuous variable into a number of bins allows its representation with a finite alphabet of symbols. This allows the use of known compression algorithms [10] to decrease the number of bits needed to represent a given attribute at the price of introducing a certain level of distortion. We employ uniform quantization where the range of a continuous attribute x is partitioned into Q equal subintervals, and all numbers falling in a given interval are represented by its midpoint. Denoting quantizer operation as $x_q=Q(x)$, the *quantization error*, $\varepsilon_q=x_q-x$, can be considered as uniformly distributed in a range $[-q/2,\ q/2]$, where $q = (x_{max} - x_{min})/Q$ is the *quantization interval*. Therefore, quantization error variance equals $q^2/12$.

Given a data vector $\{x^1, \ldots, x^n\}$ over a finite Q-ary alphabet $A=\{a_1, \ldots, a_Q\}$ (obtained by quantization), we apply Huffman coding where more frequent symbols are assigned shorter encoding lengths [5]. This provides the optimal binary representation of each symbol of A without any loss of information, such that the output bitstring length $\sum f_i l_i$ is minimized, where f_i is frequency of a_i and l_i is length of its binary representation.

4.1 Model Sensitivity Analysis for Attributes Quantization

In data reduction for knowledge discovery, preserving fidelity of all the attributes is not important by itself. A better goal is preserving the fidelity of the prediction model learned using these attributes as measured by a loss margin α_Q. With this goal, less sensitive attributes can be allowed higher distortion and, therefore, be quantized to lower resolution by using larger quantization intervals. To estimate the influence of attribute's quantization on model predictions we propose the following sensitivity analysis of a model obtained on the down-sampled data set. The outcome of this analysis allows deducing proper relative quantization levels for all attributes resulting in an efficient quantization procedure.

For a small quantization interval q_i the function $f(\mathbf{x}_{qi},\beta(n_{min}))$ can be approximated as

$$f\left(\mathbf{x}_{qi}, \beta(n_{\min})\right) \approx f\left(\mathbf{x}, \beta(n_{\min})\right) + \frac{\partial f\left(\mathbf{x}, \beta(n_{\min})\right)}{\partial x_i}\varepsilon_{qi}\,, \qquad \mathbf{(3)}$$

where ε_{qi} is quantization error with a uniform distribution over $[-q_i/2,\ q_i/2]$ and $\mathbf{x}_{qi}$ denotes an input vector with quantized attribute X_i,.

From (3) the relative MSE increase due to quantization of attribute X_i is

$$RMSE_Q(q_i) = E\left\{\left(f(x_{qi},\beta) - f(x,\beta)\right)^2\right\} \approx \frac{{q_i}^2}{12} \int\limits_{D_{xi}} \left(\frac{\partial f(x,\beta)}{\partial x_i}\right)^2 dp(x_i), \qquad \mathbf{(4)}$$

where $p(x_i)$ is the distribution of attribute X_i.

The integral in (4) could be approximated as

$$\int_{D_{xi}} \left(\frac{\partial f(x)}{\partial x_i} \right)^2 dp(x_i) \approx \frac{1}{n_{\min}} \sum_{j=1}^{n_{\min}} \left(\frac{f\left(x^j + \delta x_i\right) - f\left(x^j\right)}{\delta x_i} \right)^2 , \tag{5}$$

where δx_i is a small number (we used $\delta x_i = \mathrm{std}(x_i)/1000$).

By X_j we denote the most important attribute with the largest $RMSE_Q(\delta x_i)$, i =1,…,K. Let us quantize attribute X_j such that the number of quantization intervals inside a standard deviation of X_j is M_j where the constant M_j is called the *quantization density*. Quantization densities of other attributes are selected such that losses of all the attributes are the same. These densities can be computed from (4) as

$$M_i = M_j \sqrt{\frac{RMSE_Q(\delta x_i)}{RMSE_Q(\delta x_j)}} = M_j \xi_i , \tag{6}$$

where $\xi_i \leq 1$ is a *correction factor* that measures the relative importance of the attributes and is the key parameter allowing an efficient quantization.

4.2 Quantization Procedure for Attributes and Target

If an attribute is nominal or already has discrete values it can be directly compressed by Huffman coding. If it is continuous, its quantization can greatly improve compression without loss of relevant knowledge.

Using correction factors ξ_i, a proper M_j needs to be estimated to satisfy a quantization loss margin α_Q. For a given M_j we calculate M_i, i=1,…,K, to quantize all K attributes. We denote a quantized version of an example **x** as $\mathbf{x}_q$.

Starting from a small M_j we should estimate true loss as $\mathrm{MSE}(\beta_q(n_{min}))/\mathrm{MSE}(\beta(n_{min}))$ and should gradually increase M_j until this ratio drops below α_Q. At each iteration of M_j this requires training a new model $f(\mathbf{x},\beta_q(n_{min}))$ with quantized attributes which could be computationally expensive. However, our experience indicates that estimating $E\{[y-f(\mathbf{x}_q,\beta(n_{min}))]^2\}$ leads to a slightly pessimistic estimation of $\mathrm{MSE}(\beta_q(n_{min}))$ which can be done by using an already existing model $f(\mathbf{x},\beta(n_{min}))$ from a down-sampling phase. Hence, to improve speed without much loss of accuracy we use $E\{[y-f(\mathbf{x}_q,\beta(n_{min}))]^2\}$ in the quantization procedure. When a proper size M_j is found, quantization densities for all continuous attributes M_i are calculated from (6) and quantized accordingly.

For classification, target compression can be very successful. If a target is continuous, we propose a representation with single or double precision, since for knowledge discovery the accuracy of target is usually more important then the accuracy of attributes. Finally, after a proper quantization of continuous attributes Huffman coding is applied to obtain an optimally compressed data set. Along with the compressed data set, a small table containing the key for Huffman decoding is saved.

5 Experimental Results

To illustrate the potential of the proposed data reduction procedure we performed experiments on 3 large data sets. The first data set corresponds to a simple regression problem, while the remaining two are well-known benchmark classification problems for knowledge discovery algorithms [7].

Normal Distribution Set

We generated a data set consisting of N=100,000 examples with 10 normally distributed attributes, x_i, i=1,…,10, and target y generated as a linear function, $y=\Sigma\beta_i x_i+\varepsilon$ for randomly chosen parameters β_i and the error term being normally distributed and containing 50% of the total variance of y. Assuming standard double precision representation, the total size of this data set is 8.8MB. We chose this set to test our down-sampling procedure *DS1*, and we used an optimal linear estimator with n_0=10, a=1.5, and loss margin set to α={0.01, 0.02, 0.05}. An extremely large compression rate of up to 1,100 times to only 8KB, with minimal model accuracy loss, was achieved as reported in **Table 1**. It is interesting to observe that almost 1/3 of the reduced data set length was used for the target representation since we intentionally decided not to compress targets due to their importance.

Table 1. Data reduction results for normal distribution data set. Here α is the prespecified loss margin, M_j is the quantization density for the most relevant attribute, *loss* is an actual accuracy loss when using reduced data for modeling, *RDS* is the reduced dataset size and *C* achieved compression rate. The original double precision representation was 8.8 MB

α	**Linear Estimator**					**Neural Network**			
	n_{min}	M_j	*loss*	*RDS*	*C*	*1.5*n_{min}	M_j	*RDS*	*C*
0.01	**1900**	10	0.007	13 KB	**680**	**4220**	8	25 KB	**350**
0.02	**1120**	8	0.009	10 KB	**880**	**2250**	6	19 KB	**460**
0.05	**420**	5	0.033	8 KB	**1100**	**1020**	4	14 KB	**630**

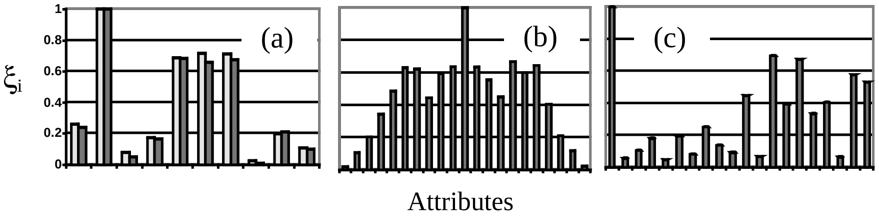

Fig. 2. Correction factors from sensitivity analysis for (a) normal distribution set (left bars are correct and right estimated correction factors), (b) `WAVEFORM` data set, and (c) `COVERTYPE` data set

We also used a neural network with 3 hidden nodes in a down-sampling procedure *DS2* to estimate the consequences of a non-optimal choice of the learning algorithm. For small sample sizes, neural networks tend to overfit the data, and hence, computed

n_{min} is significantly larger than for linear estimators. Therefore, the compression rate was slightly smaller than for a linear estimator, but still very large. It could also be noted that in both cases the quantization interval is fairly large, as could be concluded from small value of relative quantization density M_j. The experimentally estimated correction factors for 10 attributes obtained through a sensitivity analysis (right bars at Fig. 2a) were compared to the true values (left bars at Fig. 2a) and it was evident that the sensitivity analysis was extremely successful in proper estimation of attributes importance.

WAVEFORM Data Set

As originally proposed by Breiman [2, 7], we generated 100,000 examples of a data set with 21 continuous attributes and with 3 equally represented classes generated from a combination of 2 of 3 "base" waves. The total size of this data set with double precision was 17.8 MB. In a down-sampling procedure with n_0=100, a=1.5, and loss margin set to α={0.01, 0.02, 0.05} we used neural networks with 5 hidden nodes, 21 inputs and 3 outputs. Observe that the number of examples needed for successful knowledge extraction was significantly higher than in the normal distribution problem as expected for a higher complexity concept. However, for all loss margins the obtained data reduction was still very high (see **Table 2**) while estimated attributes correction factors ξ_i recovered the structure of 3 waveforms hidden in the data (see **Figure 2b)**. Our neural network models trained on a reduced data set of length 186KB achieved an average generalization accuracy of 86%, which is identical to the previously reported accuracy using all 17.6 MB of training data.

Table 2. Data reduction results for `WAVEFORM` data set (notation is same as in Table 1)

α	$1.5n_{min}$	M	RDS	C
0.01	**19670**	8	186 KB	**95**
0.02	**10640**	6	89 KB	**200**
0.05	**4580**	4	33 KB	**530**

COVERTYPE Data Set

This is currently one of the largest databases in the UCI Database Repository [7] containing 581,012 examples with 54 attributes and 7 target classes and representing the forest cover type for 30 x 30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System [1]. In its raw form it has 75.2 MB, and in the compressed 11.2 MB. Out of 54 attributes, 40 are binary columns representing soil type, 4 are binary columns representing wilderness area, and the remaining 10 are continuous topographical attributes. Seven classes represent forest cover type. Since 40 attributes for just one variable seemed too much for neural network training, we transformed them into 7 new ordered attributes by the following simple “trick”. For each of 40 soil types we calculated the relative frequency of each of 7 classes from the available examples. In that way each soil type value was represented as a 7-dimensional vector with values that could be considered continuous and were fit for use with neural networks. The transformed data set had 21 attributes and in the down-sampling procedure *DS2* with n_0=100, a=1.5, for loss margin set to α={0.01, 0.02, 0.05} we used neural networks with 5 hidden nodes, 21 inputs and 7 outputs. In the quantization procedure we quantized only 10 continuous attributes, while nominal soil type and wilderness area attributes were, together with the target variable, compressed by Huffman coding directly. Data reduction results presented in **Table 3** show that surprisingly large data reduction of several thousands times can be performed without significant knowledge loss and achieving about 70% accuracy as consistent with previous reported results [1].

Table 3. Data reduction results for COVERTYPE data set (notation is same as in Table 1)

α	$1.5n_{min}$	M	RDS	C
0.01	**6860**	8	26 KB	**2890**
0.02	**3690**	6	14 KB	**5370**
0.05	**1680**	4	6 KB	**12500**

It should be noted that approximately 1 KB of reduced data set size is used to represent a very informative 40×7 table of relative frequencies for 7 classes on 40 soil types. The estimated attribute correction factors are shown in Fig. 2c. One of the by-products of this sensitivity analysis indicates that the most important attributes for this problem are elevation and soil type, followed by wilderness area attribute.

One of the reasons for such successful reduction of this data set is possibly in its spatial component, and a relatively dense spatial grid (30×30 meters). To better exploit the spatial component of the COVERTYPE data set it would be desirable if positions of examples were also included in the form of x and y coordinates. This would allow the use of the elements of spatial statistics [3] and adjusted learning algorithms [12] for better knowledge extraction.

5 Conclusions

In this paper we proposed a set of procedures aimed at performance-controlled reduction of a given data set by: (1) elimination of redundant training examples, and (2) attributes quantization and compression. The data reduction goal was to obtain a minimal length reduced data set that, using the same learning algorithm, allows extraction of the same knowledge reduced for at most a predetermined loss margin.

Experiments were performed on a large regression and two large classification data sets. An ordinary least squares algorithm and neural networks were used to guide data reduction. Depending on prespecified loss margins of 1% to 5% of full accuracy, the achieved compression rates ranged between 95 and 12,500 times, indicating possible huge benefits for centralized knowledge discovery in distributed databases.

We obtained few other results worth mentioning. The proposed sensitivity analysis proved very successful in ranking the attributes and allowed an efficient compression of continuous attributes. This analysis can be considered separately as a method for soft feature reduction and feature selection that is based directly on their importance for a given learning model. Our results also show that a proper choice of learning model is important for data reduction and that a reduced data set can be used as a good indicator of the complexity of a learning problem.

The proposed procedure is suited for learning algorithms based on least squares minimization, and could be applied to a range of classification and regression problems. Further work is needed to extend the technique to other learning algorithms.

References

[1] Blackard, J., *Comparison of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types*, Ph.D. dissertation, Colorado State University, Fort Collins, 1998.

[2] Breiman, L., Friedman, J., Olshen, R., Stone, C., *Classification and Regression Trees*, The Wadsworth International Group, 1984.

[3] Cressie, N.A.C., *Statistics for Spatial Data*, John Wiley & Sons, Inc., New York, 1993.

[4] Davidson, R., MacKinnon, J., *Estimation and Inference in Econometrics,* Oxford University Press, New York, 1993.

[5] Huffman, D, A Method for the Construction of Minimum Redundancy Codes, *Proc. IRE*, vol. 40, pp 1098-1101, 1952.

[6] Judge, G., Lee, T.C., Hill, C., *Introduction to the Theory and Practice of Econometrics*, John Wiley & Sons, 1988.

[7] Murphy, P.M., Aha, D.W., *UCI Repository of Machine Learning Databases,* Department of Information and Computer Science, University of California, Irvine, CA, 1999.

[8] Oates, T., Jensen, D., Large Datasets Lead to Overly Complex Models: An Explanation and a Solution, *Proc. Fourth Int’l Conf. on Knowledge Discovery and Data Mining*, 1998.

[9] Provost, F., Jensen, D., Oates, T., Efficient Progressive Sampling, *Proc. Fifth Int’l Conf. on Knowledge Discovery and Data Mining*, 1999.

[10] Sayood, K., *Introduction to Data Compression*, Academic Press/Morgan Kaufmann, 1996.

[11] Stolfo, S., Prodromidis, A., Tselepis, S., Lee, W., Fan, D., Chan, P., JAM: Java Agents for Meta-learning over Distributed Databases, *Proc. Third Int’l Conf. on Knowledge Discovery and Data Mining*, 1997.

[12] Vucetic, S., Fiez, T., and Obradovic, Z., A Data Partitioning Scheme for Spatial Regression, *Proc. IEEE/INNS Int'l Conf. on Neural Networks*, No. 348, session 8.1A, 1999.