Automated Modeling and Nonlinear Axis Scaling

[SUMMARY]

Leejay Wu

Summary of dissertation to be defended:
Wean Hall 5409
May 23, 2005

School of Computer Science
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee
Christos Faloutsos, Chair; Carnegie Mellon University
Anastassia Ailamaki; Carnegie Mellon, University
Andrew Moore; Carnegie Mellon University
Richard Caruana; Cornell University
Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

This research was sponsored by the National Science Foundation under Grant No. IIS-9910606. Additional material support was provided by Intel.

Abstract

This thesis examines nonlinear axis scaling and its impact on the modeling of interattribute relationships. Through automated methods, the described system identifies possible scaling methods; decides which attributes serve as inputs or outputs; and builds regression trees that quantify these relationships. While the experiments focus on the accuracy and complexity of these models, both of which one can attempt to quantitatively examine, the results also consider applicability towards the inherently more qualitative task of rule-based outlier or anomaly detection. The results demonstrate that the use of nonlinear axis scaling, even in an automated system, can provide significantly more accurate models compared to the unscaled case without proportionally higher complexity costs; and also can help reveal unusual tuples in which what is unusual is not any individual value, but the combination thereof.

1 Introduction

This work considers data mining from the perspective of model building in an unsupervised domain, where the term 'model' specifically refers to any equation that approximates individual values of individual attributes as functions of the associated values of other attributes; and 'unsupervised' means that the system operates without requiring input-output labels or being able to request additional data. The system is also not permitted to assume that the original scales of the attributes are optimal for its purposes; for instance, it may be the case that an accurate model of acceptable complexity would be easier to build if a logarithmic transformation were applied to an attribute. To deal with this possibility, we must consider a space of possible nonlinear axis scalings and attempt to determine which will help the model-building process.

The current system may be divided into three major components: nonlinear scaling, input attribute selection, and modeling. Each stage produces output that may be of interest by itself; for instance, the choices of scalings or the division between inputs and outputs may be of separate interest. However, the modeling stage is the most amenable to quantitative analysis as we can assess the accuracy of the models more easily than we can the benefits of knowing that a particular attribute seems best scaled in a certain fashion, and is the most direct means of justifying the previous stages; therefore, the experiments and analysis thereof focus on this aspect rather than making more direct assessments of the selected axis scalings or choice of inputs and outputs.

2 Design Questions

Ideally, axis scalings, the determination of inputs and outputs, and the resulting models should be built *automatically* with a minimum of per-set tuning, so as to supplement the useful but less-scalable expertise and attention of any human domain expert. We want an fully-automated system for identifying the relationships between inputs and outputs, and for evaluating how well data fits these relationships so we can find interesting exceptions, when given a complete data set consisting primarily of numerical attributes.

2.1 Four fundamental questions

In light of the above, for any given set we might consider the following questions.

1. What scaling is appropriate for the data?
2. Which attributes should we model?
3. Which attributes should be the inputs for those models?
4. How should these models be chosen?

Of these, first examine the second and third questions. These closely-related queries stem from the fact that to model a process requires both inputs and outputs. For simplicity, let us restrict the solution space by requiring that each attribute be either an input or an output, but not both. In addition, let us suppose that our model builder can determine which attributes are most relevant among possible inputs. This reduces the second and third questions to the problem of dividing the set of attributes between inputs and outputs.

The fourth question also directly relates to the modeling problem. Even if one has been given a fixed partition between inputs and outputs, one still needs to decide how to model. The exact choice of the model class and the method for searching that specific class matters.

Lastly, consider the first of the four questions – what to do about scaling. In theory, if all one cared about were attaining precise and accurate models, many model classes possess sufficient flexibility that scaling would not matter. In practice finding optimal or even good models within sufficiently flexible model classes happens to be quite difficult; thus, scaling can affect the end result even if the model class itself contains accurate models for the unscaled data. In addition, scaling might affect the efficiency of the models. Furthermore, scaling might also have an impact on the efficacy of input selection depending on the methods used. This possibility therefore seems worth investigating.

3 Related work

Between attempts to handle the curse of dimensionality and the desire to find patterns and models, a substantial body of work proves relevant to the problems at hand. This summary will merely mention SPARTAN [2] as the single most related work to the system described herein.

4 System Design

With that in mind, consider the system as designed and implemented. The system consists of three stages – scaling, selection and modeling. Each of these stages processes the data with the intent of eventually returning efficient, accurate models demonstrating the relationships between attributes, and may return additional information regarding the nature of the data.

4.1 Axis scaling

For axis scaling, the system searches the space defined by the cross-product between two function classes: Box-Cox transformations, and the cumulative distribution functions for a variety of scalings. The system uses the former, then estimates parameters for various distributions to identify the latter; only fits surviving certain goodness-of-fit parameters are accepted. In addition, an untransformed version always survives this phase.

4.2 Input selection

The axis scaling phase results in one or more versions of each original attribute, where each version corresponds to either an identity transformation or a pairing between a particular Box-Cox transformation and a probability-based scaling. The input selection phase decides which versions of which attributes serve as inputs, discards other versions of the input attributes, and labels the remainder as possible outputs. A multi-state hill-climbing approach uses the D_2 fractal dimension to search the space of possible combinations of input attributes, while constrained by a restriction that the same original attribute never be used as an input in two different scalings. Every version of every attribute will be labeled as either an input, a possible output, or redundant due to merely being an alternate scaling of an input.

4.3 Modeling

The final stage performs modeling. The current system uses a regression tree based on SECRET [4] but with a much more aggressive pruning system, within a form of four-fold cross-validation is performed.

The question of how to fairly evaluate errors when even the scaling is in question led to the design of scaling-robust quantile-based error metric. This quantile error metric essentially involves the transformation of observations and estimated values into a percentile space; for instance, the extrema correspond to 0 and 1, the median becomes 0.5, and so forth with appropriate allowances for duplicated values. Errors computed in this space are robust to even nonlinear monotonic axis scalings, thus largely eliminating the problem of a metric giving different assessments before and after scaling for the same model.

4.4 Scalability

As with any complex system, it would be of interest to know just how long execution might take, and whether the system would scale to large data sets either as-is or with moderate modification. One might want to know what a worst-case scenario is. The described system is sufficiently complex and data-dependent to defy theoretical analysis, so this can only be estimated via experiments.

5 Experiments

The system exists primarily in Perl 5 scripts and modules as well as some C++ code. Testing proceeded on a single machine with two Intel Xeon 2.8 GHz processors and 1 GB of memory running Linux. With the relatively small sets used, processing power was far more important than memory. No attempt was made to exploit parallel computing.

5.1 The impact of nonlinear scaling

Experiments considered the following potential areas of observation.

1. What is the impact of nonlinear scaling on the attribute selection process?
2. How good are the resulting models?
3. What else can we find out?

Space constraints force the exclusion of most examples. Here follows a partial analysis of the results for one data set.

5.1.1 DJ30

The `DJ30` set is the `DJ1985-2003` submitted by Danilo Careggio to `StatLib` [3]. This set contains stock data for each stock that was a component of the Dow Jones 30 Industrials in October 2003, split-adjusted, over a period consisting of 2,529 trading days from January 2, 1985 to October 30, 2003. While one attribute was labeled `CO`, I refer to it as `KO` on the suspicion that – `CO` not having been a DJIA component while `KO` was – this was a labeling error. The truth of the matter has few ramifications other than the fact that `CO` and `KO` operated in different markets and presumably have different dependencies. The data set description indicates that the data was originally collected from `Yahoo!`.

From this set, I used the split-adjusted closing prices. While the sequential nature of the data would have allowed it, no experimentation was performed with exploiting temporal correlations. For instance, a model for the closing price of American Express (`AXP`) did not use any historical prices for American Express, and could use only the closing prices of the other Dow Jones 30 components for that same day. One could obviously perform considerable experimentation attempting to predict thirty concurrent and non-independent time series.

Without nonlinear scaling, only a single stock was chosen as an input – IP, or International Paper. Most of the quantile errors are on the order of 100 to 200, with a few notable exceptions. The least occured with DuPont (`DD`); there, a 5-node tree had a maximum quantile error of 0.48280, and a median quantile error of 7.3211×10^{-2}. The model for Caterpillar (`CAT`) is almost as accurate, with median and maximum quantile errors of 9.2970×10^{-2} and 0.49715 respectively.

With nonlinear scaling enabled, the system selects two attributes – International Paper, with a third-root Box-Cox and a truncated normal; and JP Morgan, with a mixture of normals. In all 28 models common to both the scaled and unscaled cases, the sum-of-squared quantile error has dropped. The median of the median quantile errors for all the models is down from 0.14890 to 7.4698×10^{-2}; the maximum median quantile error dropped from 0.20833 to 0.14980, with these maximum medians belonging to `WMT` in both the scaled and unscaled cases.

Outliers The nature of the data set and the accuracy of the models in the case of nonlinear scaling allows one to not only identify apparent outliers – prices for which the models have predicted unusually badly – but to search for reasons why. For one model, `CAT`, 8 of the 13 worst errors occur on the eight trading days from April 9, 2003 to April 22, 2003. All of these errors exceed 0.3700 in quantile space; the median was 7.3458×10^{-2}. It may be noted that in early April S&P downgraded `CAT` to 2 stars (avoid) from 3 stars (hold), labeling its rise in share price an overreaction [1]. The two worst quantile errors for `SBC`, of magnitudes 0.5077 and 0.5362, fall on the 20th and 21st of September, 2001, and vastly exceed the third worst error of magnitude 0.3551. On the 21st, it had announced plans to purchase outstanding shares of Prodigy Communications; this may have relevance.

SBC Communications SBC Communications (`SBC`) offers an interesting example of the improvements made possible by nonlinear scaling. With nonlineare scaling, the median quantile error has dropped from 0.16600 to 5.2031×10^{-2}. On the other hand, the scaled model shown in Figure 1 requires two inputs and eleven nodes instead of one input and one node.

Figure 2 shows the closing prices of `IP` and `SBC`, unscaled. A cursory examination shows that `SBC`'s closing price must depend on something other than `IP`; since the unscaled case resulted in selecting only `IP` as an input, `SBC` could *not* have been modeled well. It therefore becomes natural to ask whether or not the inclusion of `JPM` would suffice. Figure 3 shows the outcome, should one force this selection and build a regression tree with these inputs and output; the resulting 5-node tree has a median quantile error of 7.1377×10^{-2} and a sum-of-squared quantile error of 30.619; smaller but less accurate than the model derived with nonlinear scaling.

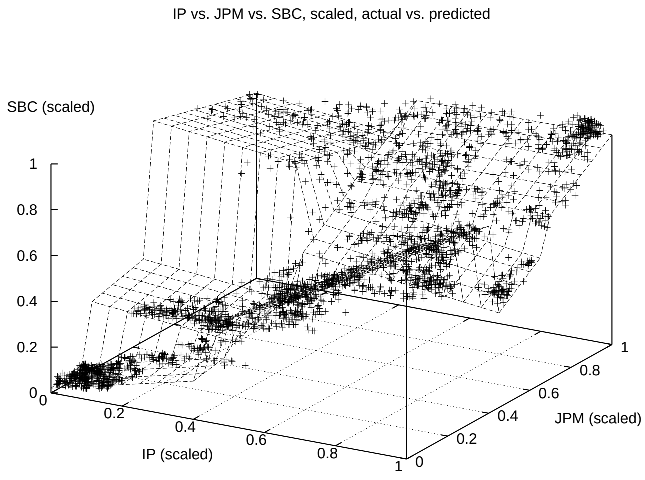

Figure 1: `IP` and `JPM` versus `SBC`, scaled. The observations, marked by crosses, are well-tracked by the model.

5.2 How long did it take?

This is really two questions: how long did it take for each set in total, and how well does the system scale with the cardinality of a data set. For brevity's sake, only the latter question is considered here.

For each sample fraction from 10% to 80% at 10% intervals, five independent samples were drawn. These samples were then processed with and without nonlinear axis scaling. Figures 4 and 5 shows the results for each individual selection, scaling or modeling phase as well as total times with and without nonlinear scaling. In both cases, CPU time required scales linearly with sample size and is dominated by the modeling phase.

6 Discussion

Recall that the major concerns of the experiments were the following.

1. What is the impact of nonlinear scaling on the attribute selection process?

2. How good are the resulting models?

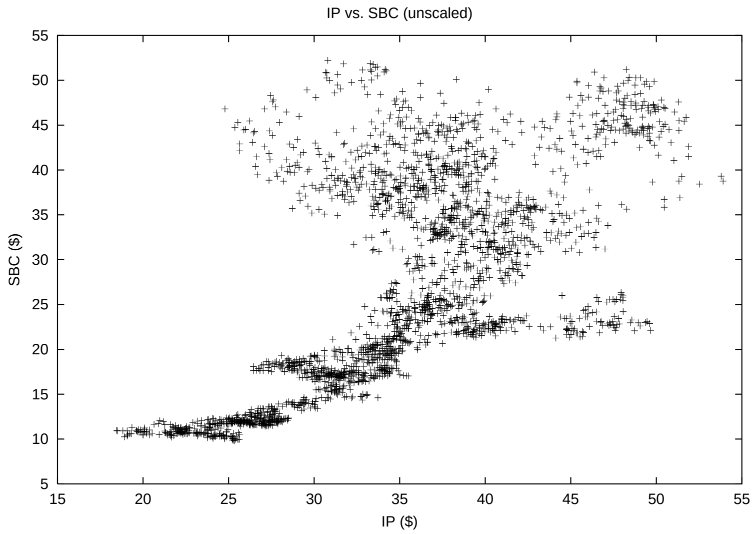

Figure 2: `IP` versus `SBC`, unscaled. Clearly one needs more than `IP` to model `SBC`.

3. What else can we find out?

The first two points encompass the major point of this project, which was to quantitatively evaluate the impact of a particular automated nonlinear scaling system and its heuristics on the overall behavior of a modeling system. Attribute selection came into play in order to enable dealing with cases in which attributes have not already been labeled as inputs or outputs. The third point deals with the possibility of other information noted either from or about the system.

Results described in the previous selection lead this investigator to a few basic conclusions. First, nonlinear scaling generally led the attribute selector to select slightly more attributes – although not necessarily an exact superset, nor that many more. Two, model performance as measured by the quantile error often improved significantly, while model complexity remained fairly stable in most cases. Third, there were some interesting trends such as the dominance of the mixture-of-normals distribution and the odd anomaly.

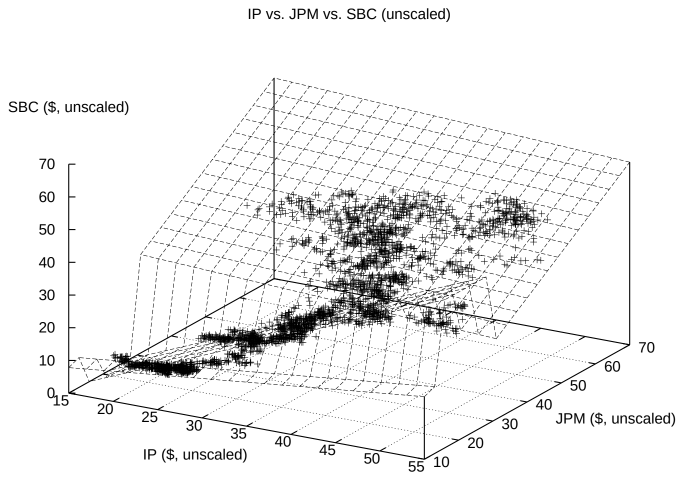

Figure 3: `IP` and `JPM` versus `SBC`, unscaled. This model is fairly accurate, but not as accurate as that of the scaled case shown by Figure 1.

6.1 Impact on attribute selection

In the experiments conducted, with nonlinear scaling enabled the selector module always chose at least as many attributes as without nonlinear scaling. It did not, however, necessarily choose a superset, nor did it ever choose many more attributes than in the unscaled case. Furthermore, should any improvements in model accuracy be credited to the selection of more inputs, that latter selection can itself be ascribed to the use of nonlinear scaling.

6.2 Impact on the resulting models

While the scaling and selection may itself provide some useful information, the construction and evaluation of regression trees allows the prospective user to evaluate the overall system through the error metric of his choice. In addition, the models themselves may be useful in their own right or the exceptions to them could be of interest; however, the utility of a model or its exceptions does not readily lend itself towards quantitative assessment. Instead, material such as the discovery of explainable anomalies consists primarily of anecdotal evidence and is relegated to the next section of this chapter.

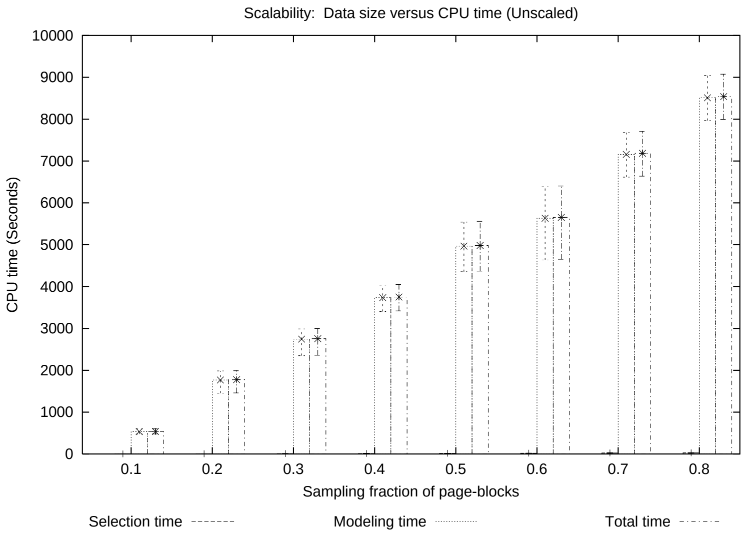

Figure 4: Timing runs at different sampling frequences over the `page-blocks` set, without nonlinear axis scaling. Error bars indicate minimum and maximum timings. For each sampling size, the bars reflect selection time – basically nothing compared to modeling time – modeling time, and total time. The total time appears to scale linearly with sample size.

That noted, regarding the performance of the modeling system I have presented three basic types of statistics where applicable. These concern the size of the regression trees, the errors produced by the trees when evaluated against the data sets, and classification accuracy where appropriate. Table 1 aggregates high-level results for the first two areas, counting attributes modeled in both the unscaled and scaled cases according to whether the models were more accurate according to the sum-of-squared quantile error, and whether they were more efficient in terms of the number of nodes in the regression trees. The 41 models which either have superior accuracy with the same or fewer nodes, or which have fewer nodes at the same or better accuracy are improvements, modulo the computational complexity of the scaling itself; the five models which are either of inferior accuracy or worse efficiency and without improvements in the other aspect are clearly worse. For four models, neither efficiency nor accuracy has changed much. The 25 remaining models with either better accuracy and worse efficiency or vice-versa would need to be considered more carefully in order to fairly

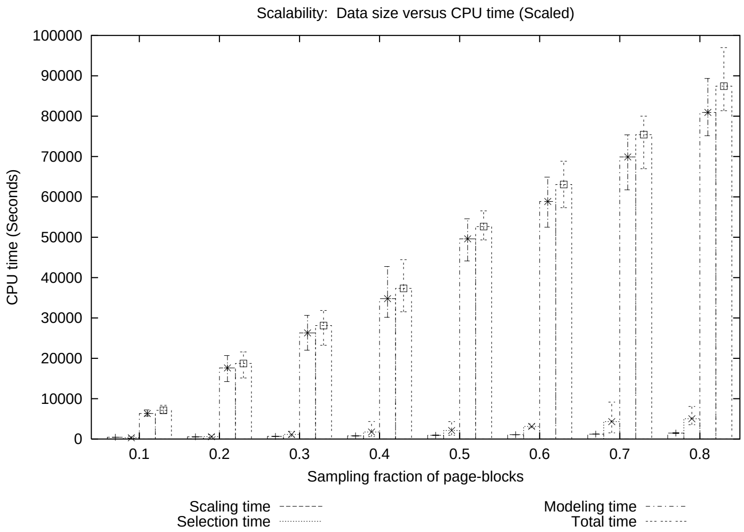

Figure 5: Timing runs at different sampling frequences over the `page-blocks` set, with nonlinear axis scaling. Error bars indicate minimum and maximum timings. Again, individual bars reflect scaling, selection, modeling and total time. The total time shows a linear correlation with sampling size.

assess their merit.

6.2.1 Model size

The first class consists of the number of nodes in the selected binary regression tree, from which one may deduce the number of test nodes and leaf nodes. The storage cost of a regression tree as implemented is currently dominated by the leaf node cost. Each test node stores a single attribute index indicating the test attribute and either a single value (for a numerical test; almost all tests were thus) or a set of symbolic values (for a membership test). Every leaf node stores a linear equation containing an additive constant and one coefficient per input attribute. The cost of the tree is conditionally independent of the use of nonlinear scaling given the tree structure, the number of attributes selected, and the number of attributes available for selection.

In practice, the majority of trees selected had 11 nodes or fewer, which means 6 or fewer linear equations and 5 or fewer test nodes. Storage costs

	Better $\sum Q^2$		**Same $\sum Q^2$**		**Worse $\sum Q^2$**	
Fewer nodes	**total**	**11**	**total**	**0**	**total**	2
	`abalone`	1			`abalone`	1
	`DJ30`	6			`building`	1
	`housing`	1				
	`page-blocks`	3				
Same # nodes	**total**	**30**	**total**	**4**	**total**	**4**
	`baseball`	11	`liver`	2	`abalone`	1
	`CIA-WF`	1	`wind`	2	`baseball`	2
	`DJ30`	7			`building`	1
	`housing`	6				
	`liver`	1				
	`wind`	4				
More nodes	**total**	**23**	**total**	**1**	**total**	**0**
	`building`	2	`abalone`	1		
	`DJ30`	15				
	`housing`	1				
	`page-blocks`	3				
	`wind`	2				

Table 1: Tallies for all attributes modeled in both the unscaled and scaled cases. The columns indicate whether or not the version with nonlinear scaling has better, approximately the same (5% difference), or worse results; the rows, whether the model after scaling has fewer, the same, or more nodes than in the unscaled case. The results are shown in both totals and per data set.

would therefore be quite small as befits small data sets, on which a larger tree would risk overfitting. Also of note is that scaling as applied rarely had much of an effect on the number of nodes. Any benefit or penalty in modeling accuracy, therefore, could not broadly be attributed to a substantial change in the model complexity other than any additional metadata defining the actual scaling. There were exceptions, such as the `area` attribute in the `page-blocks` set. That attribute received an inaccurate 3-node tree without nonlinear scaling, and a very precise 23-node tree with.

More directly related to tree size than the hypothetical use of nonlinear scaling would be the regression tree building algorithm; in particular, the splitting and pruning criteria. The system as implemented uses a particular set of rules intended to have fairly compact trees; other users may have different opinions on an appropriate balancing point. These decisions, however, were held constant regardless of the use of scaling or the choice of data set and while they introduce the possibility of differing results with other systems, within the scope of these tests the use of nonlinear scaling or the lack thereof is the single major experimental variable.

6.2.2 Errors

Without explicitly presenting the trees, data, source code and per-tuple errors, I have endeavoured to present an acceptable description and analysis of how well the models performed in their most fundamental task of regression.

Where attributes have contained only integer values, a cost model has been applied that considers the cost of both the model – using rather basic accounting such as assuming IEEE 64-bit double-precision floating-point numbers for some fields and ordinary prefix-free codes for others – and any additional storage necessary for identifying both the tuple index and the magnitude of any errors. This methodology comes from the semantic compression point of view, and allows one to balance storage cost and model accuracy in accordance with a well-known practice. Few sets used had many integer attributes, however; the two exceptions are `baseball` and `page-blocks`. As the full document notes, bit costs appear to have been reduced somewhat for those sets.

With continuous attributes, the task of fairly measuring model performance becomes considerably more complicated. I have presented a quantile-based metric designed to look solely at model accuracy from the point of view of a disambiguation task, and which therefore varies expected accuracy with the local data density. Greater precision is necessary in ranges in which more tuples fall, and less for sparser regions.

Results on the real data have also included median and maximum magnitudes of quantile errors, and the occasional note as to the possible cause of apparent anomalies. In general, the reported quantile errors are better with nonlinear scaling than without, and any increase in the tree size has usually been fairly limited. The occasional exception such as the `length` attribute in the `page-blocks` set shows itself.

6.3 Other findings

Results on data sets not described in this version of the document include a number of cases in which nonlinear axis scaling has helped produce more accurate models for which the most inaccurate tuples may be termed exceptional outliers. In some cases, the nature of the data set permits speculation as to the cause of model inaccuracy; for instance, the models for individual stock prices based solely on the prices of others in the same index could not be expected to predict acquisitions and rumors thereof that could have a significant impact on the perceived value of individual companies. The abbreviated results for the stock data notes a few instances.

A separate issue from correctness is performance. Timing tests suggest that in the current implementation, the modeling phase requires by far the most CPU time of any segment in the system, and that nonlinear scaling greatly increased the time spent there. Any effort spent on improving speed would be best directed at this phase. The timing results also suggest that merely knowing the dimensions of the original data set would not suffice to accurately estimate the required time; note, for instance, the drastic difference in required time for the similarly sized `abalone`, `page-blocks` and `wind` sets. In addition, the CPU time required appears to scale linearly with the number of tuples in the data set, at least on samples of `page-blocks`, in both the scaled and unscaled cases.

7 Conclusions

This thesis involved the design, implementation, and examination of a system for the nonlinear scaling of axes; the labeling of attributes as either inputs or outputs; and the modeling of the mathematical relationships between inputs and outputs, all in an automated fashion with no per-set tuning. Section 5 describes the experiments used to analyze the performance of this system in terms of both the accuracy and complexity of the resulting models, as well as the scalability of the system, and Section 6 summarizes the results.

As Table 1 makes clear, the application of nonlinear axis scaling has indeed helped. The use of scaling enabled more accurate models with the same or less number of nodes in the regression tree 41 out of 75 times; another 23 times, the use of scaling resulted in a more accurate but larger regression tree. In two cases, scaling resulted in the opposite trade-off of a less accurate but smaller model. Four attributes resulted in ties, with trees of the same size and essentially unchanged accuracy. Only in five cases did the system do worse in either accuracy or complexity with no improvement in the other.

The generation of more accurate models also revealed outliers. For instance, axis scaling certainly helped the models in `DJ30`, which in turn revealed a variety of periods of anomalous behavior in individual equities; anomalous behavior that, armed with dates and a search engine, could frequently be linked to events that might plausibly influence the prices in question and cause an individual stock to break away from the usual patterns. With inaccurate models, too many tuples look poor to justify searching for exceptions; but nonlinear axis scaling, even when automated without domain knowledge or data-specific tuning, can improve those models to the point that interesting exceptions can emerge.

In short, nonlinear axis scaling can help – a lot. It's not *guaranteed* to do so, and it adds significantly to computational cost, but it's a powerful tool and one worth serious consideration in addition to simpler methods such as affine transformations solely for normalization.

References

[1] BusinessWeek Online, April 2003. `http://www.businessweek.com/investor/content/apr2003/pi2003047_3105_pi010.htm`.

[2] Shivnath Babu, Minos Garofalakis, Rajeev Rastogi, and Avi Silberschatz. Model-based semantic compression for network-data tables. In *Proc. of NRDM 2001*, May 2001.

[3] Danilo Careggio. DJ 1985-2003. `http://lib.stat.cmu.edu`.

[4] Alin Dobra and Johannes Gehrke. SECRET: a scalable linear regression tree algorithm. In *Proc. of ACM SIGKDD*, pages 481–487, July 2002.