

Case Studies in High-Dimensional Classification

C. Apte, R. Sasisekharan, V. Seshadri, and S.M. Weiss

Journal of Applied Artificial Intelligence

Vol. 4, No. 3, July 1994

Contents

1	Introduction	2
2	Case Study 1: Disk Drive Assembly Quality Control	3
2.1	Methods and Procedures	6
2.2	Comparative Results for Alternative Learning Models	6
2.3	Results for Faulty Disk Drive Detection	8
2.4	Results for Improving Disk Drive Assembly Throughput	10
2.5	Significance of Results for Disk Drive Assembly Quality Control	12
3	Case Study 2: Finding Chronic Problems in Large-Scale Communications Networks	14
3.1	Methods and Procedures	16
3.2	Comparative Results for Alternative Learning Models	18
3.3	Results for Predicting Chronic Circuit Problems	19
3.4	Significance of Results for Predicting Chronic Network Faults	20
4	Concluding Remarks	21

Case Studies in High-Dimensional Classification

Chidanand Apté
IBM T.J. Watson Research Center
Raguram Sasisekharan, V. Seshadri
AT&T Bell Laboratories

Sholom M. Weiss¹
Rutgers University

Abstract

We consider the application of several compute-intensive classification techniques to two significant real-world applications: disk drive manufacturing quality control and the detection of chronic problems in large scale communication networks. These applications are characterized by very high dimensions, with hundreds of features or tens of thousands of cases. The results of several learning techniques are compared, including linear discriminants, nearest neighbor methods, decision rules, decision trees, and neural nets. Both the applications described in this paper are good candidates for rule-based solutions because humans currently resolve these problems, and explanations are critical to determining the causes of faults. While several learning techniques achieved competitive results, decision rules were most effective for these applications. It is demonstrated that decision (production) rule induction is practical in high dimensions, providing strong results and insightful explanations.

1 Introduction

We consider the application of several compute-intensive classification techniques [14] to two important real-world applications: (a) disk drive manufacturing quality control and (b) chronic problem detection in large scale communication networks. Unlike many applications often reported in the research literature, these applications are characterized by very high dimensions, with hundreds of features, and tens of thousands of cases. We examine the efficacy of modern search-based classification techniques to select dynamically the right features for classification when most features are

¹This research was performed while the author was a visiting researcher at IBM T.J. Watson Research Center and AT&T Bell Labs.

poorly predictive, and the prevalence of one class overwhelmingly dominates the others.

Our first effort in this area supplements an existing expert system, RAES, that is currently deployed on a disk drive manufacturing line. Many staff-years have been expended to select the right tests and to improve that system to its current expert performance level. In addition to the RAES knowledge base, a wealth of empirical information is available in the form of stored records of manufactured disk drives. In our second application, we examine the complete AT&T network to see whether patterns can be found in circuit faults that go undetected for relatively long periods of time. With the increasing computational power of generally available computers, we have seen a resurgence of interest in automated methods that learn from data, such as neural nets and decision trees [16].

2 Case Study 1: Disk Drive Assembly Quality Control

Given the current high performance of the RAES system and the engineers' strong knowledge of the area, our efforts were not directed towards replacement of the expert system. Rather, our efforts were a search for knowledge that would assist the engineers in providing another increment of performance to the already highly performing knowledge-based system. Such knowledge could only be obtained by intensively exploring high-dimensional data: the huge volumes of records of disk drive testing and performance that are recorded during the manufacturing process.

We first outline the manufacturing of disk drives. At various points in the manufacturing process, three phases of tests are performed on each drive. These tests produce numerous recorded measurements, most of which are recorded as continuous values. A smaller group of measurements can be characterized as categorical values, and they are recorded as true or false.

The first phase of testing is performed before the frame electronics card is added to the drive. Tests in this phase measure the performance and placement of the components in the drive thus far, especially in regards to the reading and writing of information on the disk surfaces.

The next phase of testing is performed with the frame electronics installed. The tests in this phase measure, among other things, the ability of the frame electronics to interface with and control the read/write mechanisms. Each of the tests in these first two phases requires relatively little

time to complete.

Once a drive has passed all tests in these first two phases, the third phase begins with a very lengthy test called RunIn. The purpose of this test is to run the drive vigorously for a long period of time in hopes of catching the majority of early-life failures before shipment to a customer. Though RunIn itself does not have great direct expense per drive, it causes inventory build-up, can be a bottleneck in the test process, requires a large number of testers to handle the volume of drives, and greatly increases the cycle time per drive. Because of these things, RunIn is very expensive and most desirable to eliminate. Figure 1 approximately illustrates the testing scenario at the disk drive manufacturing site.

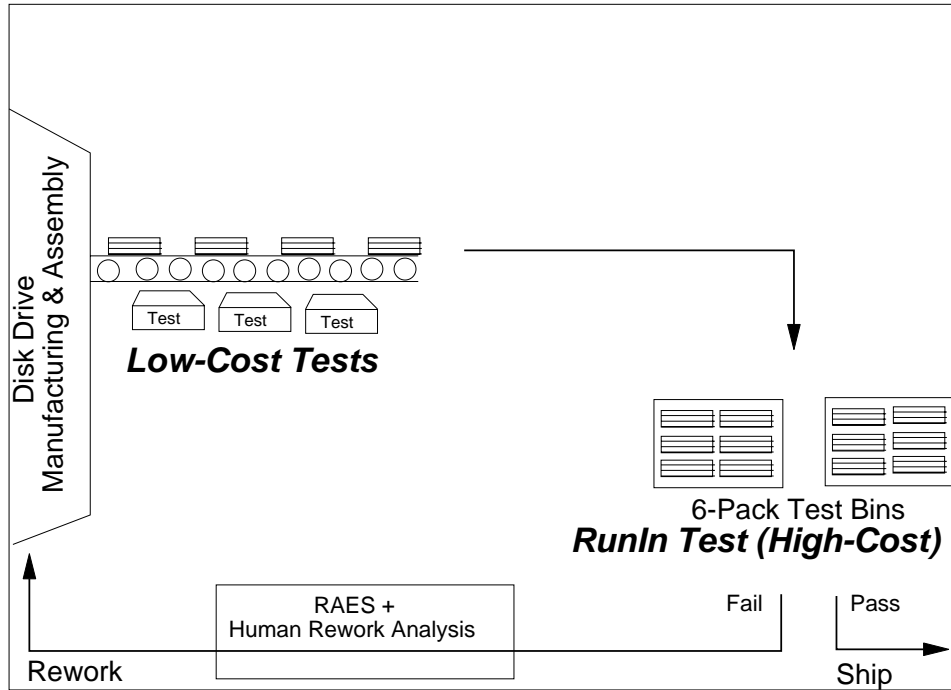


Figure 1: Approximate Current Test Scenario

The RunIn test indicates a pass or fail for a disk drive and also produces several hundred numerical measurements. The results of RunIn are used in determining whether the disk-drive is ready to be shipped. If the RunIn results indicate failure, an expert system, RAES, is used in the majority of cases to diagnose the rework that will be required to fix the malfunctioning

drive. RAES selects some rework action, usually the replacement of one of a number of components, which it determines will fix the problem. RAES arrives at its conclusion mainly by analysis of the measurements taken at RunIn and relevant previous tests. In some cases, RAES does not have the knowledge required for diagnosis, so a human rework analyzer makes the determination. A bad drive is sent back to be fixed. A reworked drive goes back through part or all of the assembly and test process, including one or more short tests and the long-running RunIn test. A relatively small percentage of the drives fail RunIn. RAES and the rework analyzers together predict the correct rework somewhat less than 100% of the time.

The total number of measurements conducted for each drive (excluding RunIn measurements, but including the outcome of RunIn and the rework recommendations) number well over 600. In terms of data size, thousands of bytes of data are captured for each disk drive that comes off the line. Given the high volume manufacturing capacity of the site, and the fact that these data were being gathered meticulously, there is a huge volume of available data. This paper describes our experiences with applying high-dimensional classification techniques to this data set in an attempt to induce knowledge that could be used for improving the manufacturing process.

The objective of this project is to improve the manufacturing process by predicting the failure of a disk drive prior to the expensive RunIn test. All predictions will be based solely on the results of three of the most relevant tests that occur prior to RunIn. Two types of improvements have been identified:

1. The elimination of the RunIn test for all or some disk drives.
2. The division of the disk drives into two groups, one of which consists of disk drives that are likely to be faulty.

These objectives can be restated in terms of specific classification and prediction problems:

1. Can both the results of the RunIn test and the diagnosis by RAES or a human be predicted for all cases? This task mandates the prediction of the failure of Runin and the determination of one of six classes of failure.
2. Can the results of RunIn be predicted without making a direct determination of the cause of failure and the component that should be replaced?

2.1 Methods and Procedures

The initial sample consisted of 36,294 records, most of which represent good drives, and the rest drives that needed rework. The result of RunIn is recorded as true (pass) or false (fail). Also recorded in each sample is the rework action recommended for repairing the disk drive. The recommendation is a selection of one of six classes, corresponding to five possible component replacements and one miscellaneous reworks category. While it is desirable to have an objective measure of truth, the only available measure of truth is the classification by RAES and the rework analyzers, which is less than 100% accurate.

The complete elimination of RunIn requires the prediction of the rework diagnosis, a 7 class classification problem: a good disk drive or one of six bad disk drive types. The prediction of RunIn alone, i.e. the prediction of bad disk drives without identifying the type of problem, is a two class problem. A solution to the RunIn prediction problem may yield efficiencies in performing the RunIn tests. However, the elimination of RunIn requires stronger prediction capabilities than the division into good and bad disk drives.

Two populations were considered. N2 is the true population of 36,294 cases. N1 consists of a smaller number of records divided almost equally between good and bad drives. The N1 population has a proportion of normals that is significantly less than the true proportion. Results on the N1 population are much more easily obtainable, but will be overly optimistic because each error on the normals may actually represent many more errors in a real scenario. However, results on the smaller N1 population will measure the potential for complete success on the true population. Poor performance on the smaller sample set can readily be extrapolated to the true population. For each sample, 564 measurements were obtained and 7 classes were considered. These features were selected from the original 600+ features by simply eliminating all constant value fields.

2.2 Comparative Results for Alternative Learning Models

Rather than concentrate on any single method, we favored a balanced approach that applies several well-known learning methods in their standard classical form [12]. If any method demonstrates a clear superiority over the others, further experiments can be performed to elicit the best performance of that method. For our quality control application, if no method is clearly

k	Error Rate
1	.52
5	.49
11	.48
25	.48

Table 1: Results for k-Nearest Neighbor

superior, the rule-based solutions are particularly advantageous. They have an inherent explanatory capability that is most suitable for further discussions with the manufacturing engineers. Because the problem is one of very high dimensions, even with high performance workstations, timing considerations make it necessary to emphasize only the most promising directions.

Five classification methods were tried: Linear Discriminant [7], k-Nearest Neighbor [4], Neural Network [8], Tree Classification [2], and Rule Induction [3, 9, 10, 11, 15]. These methods were applied to the smaller N1 population. All error rates were measured by test cases obtained by randomly holding out 1/3 of the sample cases. No method achieved an error rate better than 38%. The following are the results for the (reduced size) N1 population.

k-Nearest Neighbor

The feature values were normalized by means and standard deviations, Euclidean distance. After the method demonstrated poor results on the original set of 564 features, the set was reduced to a smaller set of 223 features that showed some significance as measured by standard statistical tests. In Table 1, we list the results for k-nearest neighbor, where k is varied from 1 to 25.

Linear discriminant (parametric) with feature selection

We could not obtain a discriminant for the full 7 class problem. For class 5, the largest class, versus all other classes the error rate is .40 with 75 features selected.

Decision trees (CART)

The final tree has only 3 terminal nodes. The estimated error rate is 41%. (A ten-fold crossvalidation estimates the error rate at 39%.) If the proportion

for each class is adjusted to its correct value, no tree is induced and all samples get classified into the first category (good disk drive, perform RunIn and ship).

Neural Nets (Standard Backpropagation)

Feed-forward, fully-connected, neural networks were trained. These are the standard back-propagation networks. A single hidden layer was used with hidden units varying from 0 to 20 hidden units. The best generalization was achieved at 0 hidden units with a 38.5% error rate on the test cases.

Rule Induction

The Swap-1 [15] procedure for rule induction was applied to the data. The result was a simple rule of the form $X > a$ or $Y > b$ or ..., having an estimated error rate of 39%.

The results of all classification methods were very far from our objective of perfect classification. However, the results did hint at a limited predictive capability for some of the tests. We therefore reconsidered a variation of the rule induction approach to find partitioned subpopulations where good predictive performance was possible.

A set of production rules were induced that were trained on the (smaller) N1 population. These decision rules make no errors on either the N1 or the full (36,294) N2 populations. They select a fairly small portion of the bad disk drives, but do not make a decision on the other disk drives. Thus on a very small group there were hints that it may be possible to avoid RunIn. Further testing on new data was necessary to validate this hypothesis. We obtained an independent set of new data. This new sample was from a more recent snapshot of the assembly line and consisted of 51,047 cases. The original set of rules were based on a very small sample. When tested on the new sample, the rules did not perform perfectly, eliminating their feasibility as a good predictor.

2.3 Results for Faulty Disk Drive Detection

Based upon the results of the rule induction techniques, we hypothesized that bad disk drives could be identified by concentrating on only a few key fields, from amongst the 600+ features that are present, that have a strong presence in the rule set. Fields 595 through 608 have been identified as the key fields. We also hypothesized an artificial test that is a derivative of

X	Predictive Value	
	Original Sample	New Sample
0.50	0.859	0.831
0.75	0.913	0.860
1.00	0.950	0.862
1.25	0.945	0.856
1.50	0.956	0.877
1.75	0.944	0.893

Table 2: Predictive Value Analysis

other tests: the number of these key fields that have a value of 0.05 or more. Analogous to rule induction techniques for medical diagnostic testing [5], we considered thresholding these tests to determine the predictive value at each numerical cutoff, a process known as referent value analysis. In purely symbolic terms a single best rule has been induced in the following form:

If

5 or more fields in 595-608 are ≥ 0.05

OR

There exists a field in 595-608 that is $\geq X$

Then

The drive is bad with probability PV.

X was varied over a range and the corresponding PV computed for this rule over all 36,294 samples.

Because predictions are made with an extremely small feature subset on a large population of 36,294 cases, it was conjectured to be highly likely that this result would hold for future predictions. To continue the verification of this hypothesis, it was tested against the second sample that was obtained from the disk drive manufacturing facility. This data set contained 51,047 samples, of which again a small portion were failures. The ratio of failed disk drives as a proportion of the total output is approximately the same as that of the original sample that we worked with. We then applied referent value analysis using our most promising rule on this new data.

Comparing the results of our predictive value analysis on the original sample set and the new sample set (Table 2), we can observe that the strength of the predictive values seem to remain consistent, with some very minor deviation. We combined both the new and old sample sets and applied the rule to the combined set for a range of X values, to compare the

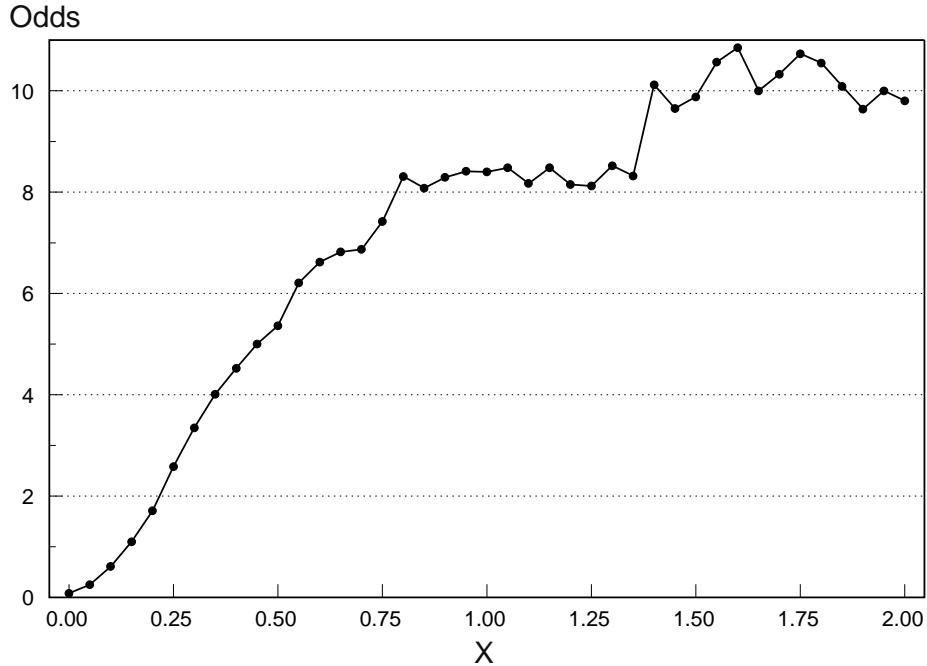


Figure 2: Odds of Detecting Bad Disk Drive in Combined Sample Set

odds that a detected disk drive is bad and the corresponding fraction of total number of bad disk drives that are detected. These results are plotted in Figures 2 and 3. These plots illustrate, for example, that if X is chosen to be 0.6, then the odds of detecting a bad disk drive are approximately 6:1, and that about 10% of the total bad disk drives will be detected.

2.4 Results for Improving Disk Drive Assembly Throughput

In this section, we review one possible approach to improve manufacturing efficiency using the rule for faulty disk drive detection. The RunIn test is a particular bottleneck in the assembly process. When a disk drive is ready for RunIn, it is packed with others in a testing bin, six to a bin. The 6-pack bin performs RunIn on all six disk drives and is unloaded when all drives in the bin have completed their tests. A bad disk drive will finish sooner than a normal disk drive. Although the six disk drives are tested in parallel, the bin can not be unloaded until disk drive testing has been completed for all

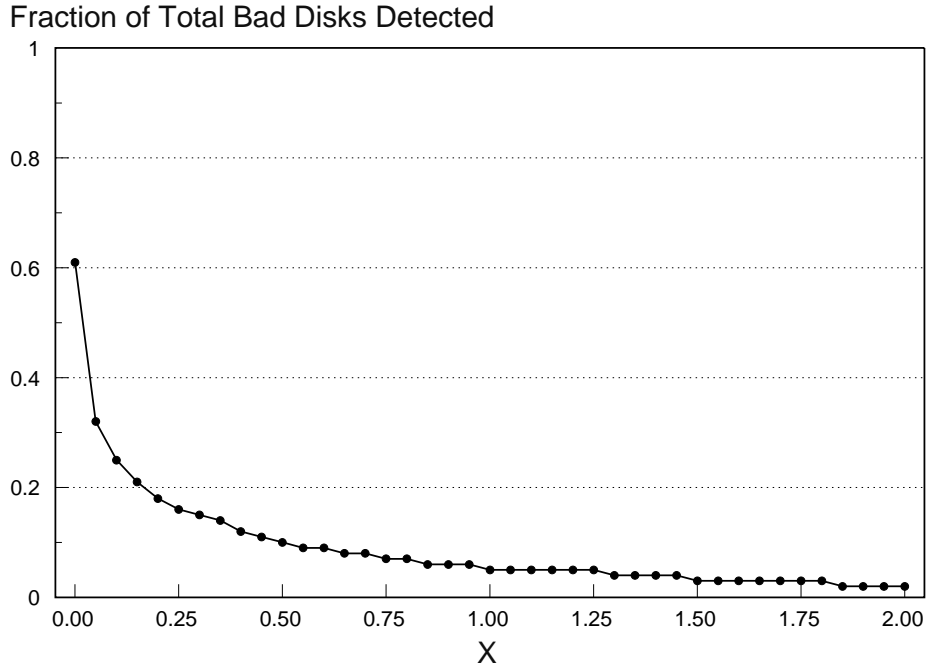


Figure 3: Bad Drives Detected as a Fraction of Actual Total Bad Disk Drives

six disk drives. If the 6-pack testers can be loaded with bad disk drives, the test process may terminate more quickly, reducing the test queue bottleneck.

Predicting the pass or failure of RunIn corresponds to the determination of a bad disk drive without identifying the failing component. Time may be saved by packing only bad disk drives in the 6-pack bin. Because a bad drive finishes RunIn much earlier than a good drive, a 6-pack loaded with only bad drives will terminate earlier than one which has a combination of good and bad drives. Currently, the probability of loading a six-pack with all bad drives is essentially 0.

Assuming that the desired predictive value is to be at least 90% (in the original sample), the proper value for X can be determined and the number of bad disk drives that may be detected can be calculated. It is likely that by using a rule of this type, the throughput of the 6-pack RunIn process can be improved. Currently, the probability of having a 6-pack bin loaded with all bad drives is essentially 0. If a 6-pack is loaded with disk drives satisfying this rule, the probability is > 0.5 that all the disk drives are bad.

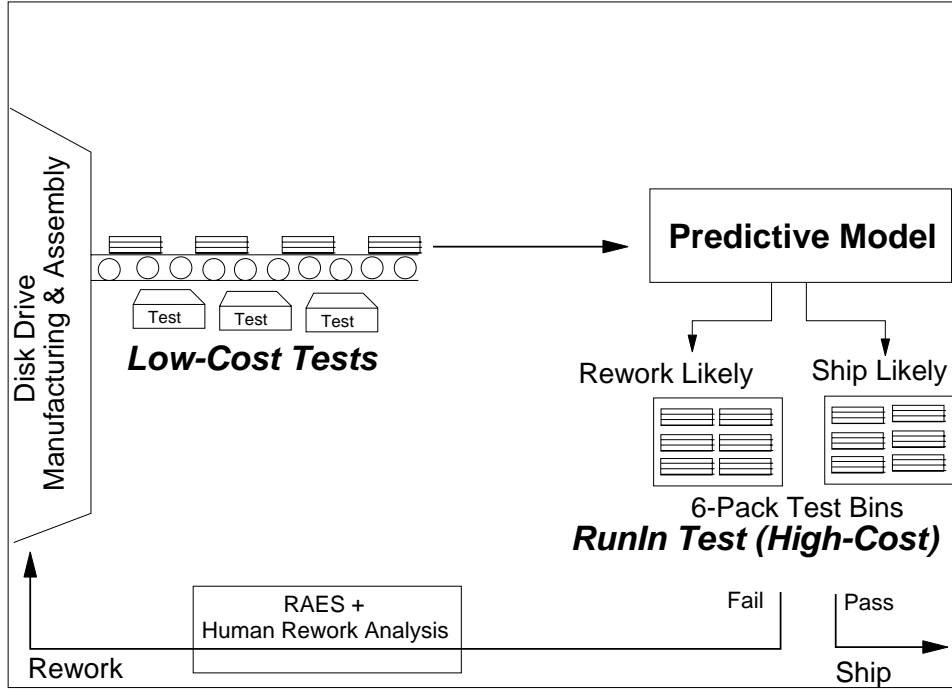


Figure 4: Proposed Test Scenario

About 8% of the bad disk drives satisfy this rule. The actual economies of disk drive screening would depend on numerous factors including the time savings, scheduling, and the number of available 6-packs. The proposed modification to the testing scenario is illustrated in Figure 4.

2.5 Significance of Results for Disk Drive Assembly Quality Control

A massive amount of data was available for performing our experiments. The two samples that we examined were about 200MB in size. They represent a small fraction of the total data. The high dimensionality of the data (over 600 features per sample) adds a degree of complexity to the analysis. The bulk of these features represent electrical, mechanical, and digital measurements. As is the case with any recordings of this nature, they are potentially prone to noise as well as calibration problems. Our current attempt has focused on combining some preliminary feature analysis with the

more widely-used classification methods.

Five classification methods were tried; Linear Discriminant, k-Nearest Neighbor, Neural Network, Tree Classification, and Rule Induction. Using these methods and the sample data and feature set, we performed numerous simulation experiments that suggest the following conclusions:

1. The elimination of the RunIn test for all or some disk drives is not feasible.
2. About 8% to 9% of the bad disk drives can be detected with greater than 80% certainty.

The first result was fully consistent with our expectations. We did not expect to find that the manufacturing engineers had needlessly been using expensive tests when in fact the inexpensive tests could provide the same information. On the other hand, the amount of human effort expended in the data analysis is minimal, and the cost of computer processing pales in comparison with typical knowledge acquisition efforts. Thus, exploring the possibility of eliminating RunIn was a reasonable investigation.

The complete elimination of the RunIn test requires perfect classification. Bad disk drives cannot be shipped so all expected normal disk drives must be tested by RunIn. Good disk drives cannot be sent back for rework because rework is even more expensive than RunIn. The experiments strongly suggest that perfect accuracy in classification cannot be achieved using solely the measurements from the three inexpensive groups of tests. If however, one wishes to pursue further experiments, future directions might include the application of disk drive manufacturing knowledge in conjunction with statistical and information theoretic methods to come up with a more meaningful feature subset. Variations on the distance metric in the k-nearest neighbor technique, or in neural network topologies are also possible alternate experiments to undertake. However, the consistent results that we obtained across an entire family of well known and powerful methods suggest that not much more may be gained without stronger features, i.e. more relevant test measurements of the disk drives during the manufacturing process.

3 Case Study 2: Finding Chronic Problems in Large-Scale Communications Networks

With the increasing size and complexity of modern communications networks, there is a commensurate need for intelligent systems to help manage and maintain them. It is desired that these systems analyze and resolve problems in the network automatically. In addition, they should identify potentially serious problems before they degrade, thus greatly improving the reliability and quality of the network. The magnitude of this task is extremely large. A network such as AT&T's world-wide network is a massively interconnected structure of a large number of complex devices with over a million different paths or circuits. During any day, billions of transmissions are made over the network. Many sophisticated computer systems and devices interact to achieve the extreme reliability that AT&T maintains over its network.

There are various aspects of managing and maintaining such a network, ranging from managing the resources and traffic to troubleshooting different kinds of problems in the network. Many computer systems have been built to automate the management and maintenance of networks. Specifically, systems have been built to automate the diagnosis and repair of transmission problems in the network. These earlier methods have concentrated on incorporating knowledge obtained from domain experts in rule-based systems to troubleshoot problems. These systems focussed on resolving hard failures that had occurred in the past. Increasingly, there is a demand for maintaining the network proactively. This includes the ability to predict problems that are likely to persist, and to predict those problems that are likely to degrade. This is a very important aspect in maintaining a high level of quality and reliability in the network.

Proactive maintenance of the network can be accomplished by monitoring the performance of the network continuously over time and identifying patterns that are indicative of future problems. Monitoring network performance involves analyzing extremely large amounts of diagnostic data that varies with time. In this part of the paper, we will describe our analysis of data from transmission problems in AT&T's world-wide network.

The Application Domain

We will first provide a brief background of the domain, starting with a communications network. A communications network can be considered to have

many terminal nodes, each of which can communicate with others through devices such as switches, multiplexers, cross-connects, etc. Network operation systems (NOS) exist in the network to support provisioning, maintenance, operation, administration and management functions for the network and for individual network components. A circuit can be considered as a path between two terminal nodes, which contains network components and links. Transmission problems on a circuit are seen by several of the network components through which the circuit connects. In a large network, such as AT&T's communications network, the ratio, of diagnostic data generated by various network components to the root problem that is responsible for them, is large.

The different types of problems in the network can be broadly categorized into two classes, transient and non-transient. Transient transmission problems are very common in the network, yet their behavior and causes are not completely understood. Part of the difficulty in understanding them is related to separating the wheat from the chaff, that is, in learning to ignore glitches that will not be repeated and focusing instead on those transient problems that will recur (chronics). Chronics not only affect the quality of communications when they recur but also indicate degradation and potential future failures in the network. Thus it is an important and challenging problem to identify these chronics and isolate their causes.

Diagnostic procedures that attempt to resolve transient problems must rely on large volumes of historical information and a more complex analysis of patterns of behavior. One novel approach to diagnosing transient faults is found in an AT&T system known as SCOUT[13]. Using historical and topological information, SCOUT finds specific related circuits that share common patterns of faulty behavior. Typically, these are difficult transient problems, multiple circuit problems, or even forms of chronic faulty behavior. In this study, we considered a related form of analysis of chronic behavior: the performance of the complete AT&T network over time. The objective is to determine whether there are patterns of behavior over the network such that it can be predicted that the faulty behavior will continue in the immediate future.

The sample size in this analysis numbers in the tens of thousands of cases. Looking for patterns of behavior in such large volumes of data can only be accomplished by computer analysis using machine learning techniques, possibly resulting in new information that cannot be obtained by typical human experience.

3.1 Methods and Procedures

Describing the Goals and Measurements

The circuit-related questions that were outlined in the previous sections need to be posed in a standard classification format, so that a number of interesting analytical techniques that are available can be applied. Prediction models that can be applied to a standard classification problem include decision trees, decision rules, statistical linear discriminants, neural nets and nearest neighbor methods. In the standard classification format, samples of cases are obtained. For each case, identical measurements are taken, and at least one of these measures is the class label. Methods are applied which attempt to find patterns for one class that differ from other classes.

For our problem, the class label is chronic failure on a circuit, a concept that has been defined in previous sections. The goal is to predict that current failures will continue to occur. We must also take into account that these failures are often transient, and that failures will likely not occur continuously in the future. Instead, a failure may occur in the future, but the occurrence may also be transient. Periods of time that are reasonably close to the current period are of interest.

The measurements that are used for prediction must summarize historical information. These measurements are recorded each time a fault occurs. Not all measurements are recorded for every fault, only those that directly measure the fault process. Faults are often transient, so the trends for a period of time must be measured. It is quite possible that many faults will occur for a short time, but these faults are not necessarily chronic. They can be fixed and do not reappear in the immediate future. Measurements must be specified that are useful in predicting the target concept, that is, future failures on the circuit.

This time-dependent problem was mapped into a standard classification format by the use of fixed time windows. Historical information for circuits was examined over a consecutive period of time, and this time period was divided into two windows, W_a and W_b . The objective was to use the measurements made in W_a to predict that problems will occur in W_b . We considered both our knowledge of the application and experimental data to arrive at reasonable sizes for each of the two windows. The windows were also divided into sub-units based on time, which we will refer to as a time unit. We will refer to the size of W_a as T_a time units, and the size of W_b as T_b time units.

There are many reasonable measurements that can be taken over time. Assuming a fault occurs, an alarm or exception is noted. Included in the possibilities of measurements are the number of times such an event occurs, the average number of times an event occurs during a time unit, or the number of time units during which the event occurs. Also included in the measurements are the subsection of the network in which the events are observed. We defined 30 performance features for this problem, based on the variation of diagnostic or telemetry data over time and over space.

For this application, a large number of samples were obtained from the operating communications network. It was not feasible to try every variation of these methods on the entire set of samples. Instead, we performed some smaller experiments to see whether one approach offered an advantage over the others. The results of these experiments are discussed below. Overall, for this application, nearest neighbor methods and statistical linear discriminants performed poorly. Neural nets and decision rules or trees were competitive, with a small edge for decision rules.

In experiments on the complete data sets, representing all circuit problems in the network over a fixed period, we relied mostly on rule induction. In our study, we emphasized rule induction[15].

It is quite possible that tuning many of the alternative methods could result in somewhat improved results for each method. However, based on our knowledge of the application, there are a number of reasons why the rule induction method appears most appropriate:

1. The objective is to extract new information from the data. The hope is that we can gain insight into the performance of the network. Decision rules have the strongest explanatory capabilities of the cited models.
2. We know in advance that this is a noisy environment. Perfect classification can be achieved on all samples only when chronic behavior is entirely consistent. This is not likely with all the efforts toward high reliability and the transient nature of many problems. Thus the expectation is to find a subset of conditions that are highly reliable predictors of chronic failure.
3. Most of the measurements are ordered discrete variables. They are not continuous. Patterns of these types of measurements are usually effectively described in terms of the greater than or less than operators which are used by the rule induction model.
4. The minimum error solution is not necessarily the best solution. Because we are trying to extract new information, the preferred solution

consists of the highest predictive rules even if they cover fewer cases. The preference is also for simple rules that enhance our understanding of network performance.

Given that the expectation is for predictions that cover only a partial number of chronic problems, decision rules most naturally model the partitioning of data. The efficacy of the individual rules can then be tested on independent test data.

Testing the Decision Model

The central method for building a predictive model is to learn from samples and test on independent data. In many applications, there is a relative shortage of data. In these situations, a compromise is made by randomly partitioning the data into training and test sets. In our application, we had a large number of samples. Training was performed on a random subset of data for a time period, and some preliminary testing was done on the remaining data. Once a solution was found, further rigorous testing was performed by testing the solution on additional data from subsequent time periods.

3.2 Comparative Results for Alternative Learning Models

In this section, we describe the results that were obtained in identifying chronic problems. We have also provided the results of our comparison of various learning methods as applied in this case.

We examined the historical records for several months during late 1992 and the first half of 1993. These samples were taken from the complete AT&T network, and covered all of the transmission problems encountered. When compared to the billions of transmissions during a month and the size of the network, the number of problems is quite small. However, from a sampling perspective, we had a large sample, consisting of tens of thousands. Of these circuits, between 5 and 10% fulfilled our definition of chronic, that is, they had faults during at least half the time units during W_b .

Although we concluded that rule induction was the preferred learning method for this application, we performed several experiments to evaluate its competitiveness with alternative learning models. Table 3 summarizes the results. One third of the cases were randomly selected for testing, and the error rates in the table are based on test case performance.

Method	Error Rate(%)
Prior	6.4
Linear discriminant	6.4
Nearest neighbor	5.6
Neural net	4.7
Decision tree	4.5
Decision rules	4.4

Table 3: Comparative Results for Alternative Learning Methods

Simply choosing the largest class gives an error rate of 6.4%. The linear discriminant (Fisher’s) that we used was the standard parametric discriminant found in statistical packages. We used it with feature selection. It is not unusual that this method does not perform well in a low prevalence situation.

Nearest neighbor methods are greatly affected by noisy variables. This result is for $k=1$ and Euclidean distance.

The three remaining techniques were relatively close for this experiment. The decision tree was induced by CART, and the decision rules by Swap-1. The neural net was a standard backpropagation network, with a single hidden layer. Configurations were considered with from 0 to 6 hidden units, with the best test error rate for 0 hidden units.

3.3 Results for Predicting Chronic Circuit Problems

We now return to the central task, namely, whether we can predict chronic problems, that is, problems that will continue in the immediate future. We were able to find rule sets that were predictive. The rule sets were able to predict over 40% of the chronic problems with over 90% accuracy. The rule set consists of combinations of conditions. If any of the conditions hold, the problem is very likely to be chronic. The conditions were of the form $\text{Feature}_i > n_i$, where Feature_i is a performance feature based on the number of time units during which an event of a type i occurs, and n_i is an integer. Figure 5 plots the performance of a highly predictive rule set over the course of several time periods.

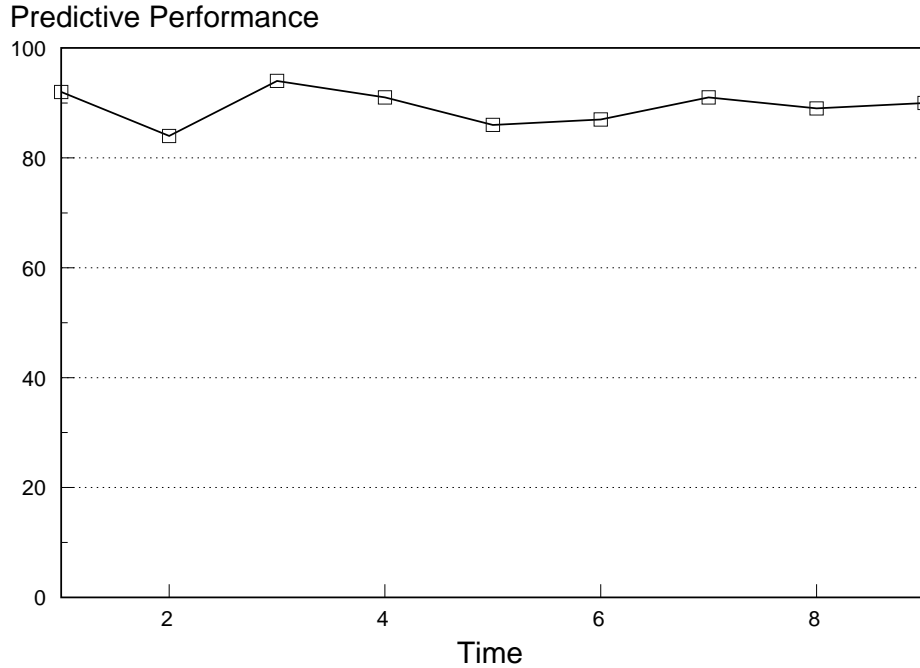


Figure 5: Predictive Performance of Rule Set Predicting Chronic Problems

3.4 Significance of Results for Predicting Chronic Network Faults

If all problems in communication networks were either transient or quickly repaired, it would not be necessary to detect chronic problems. However, chronic problems do occur. Identifying patterns of these problems is critical in characterizing problems that are not detected and repaired quickly. In our analysis, we found that the number of time units over which events occurred was significant in determining the likelihood that the problem would continue in the future. These rules suggest a form of momentum or inertia for chronic fault problems. There are a number of rationales for the validity of this form of analysis:

- Not all faults have this momentum to the same degree. We have identified those measurements that are predictive along with the corresponding thresholds, that is, the number of time units of faults during a window for which they are predictive.

- Of particular importance, any circuit that exhibits this behavior will likely continue this behavior. Thus, if the goal is to maximize reliability, circuits exhibiting these characteristics should be given priority in diagnosis and repair.

By tracking the problems that were identified as chronic into the future, that is, after they have been recognized as chronic by our method, we are able to show that these problems generated a significant percentage of error messages that are reported in the future. Thus the predictions using our method will be able to improve the performance of the network, reduce overhead in maintaining the network, reduce customer affecting incidents, and prevent potential loss of business.

4 Concluding Remarks

Increasingly, high volume domain specific data is being made available for knowledge engineering activities across a broad range of applications [1, 6]. There is great potential for using this corpus of data for analysis and induction of hidden knowledge. The high dimensionality of these large data sets will make the application of many classification techniques more complex than usual. For these applications, often with many weak or even useless features, machine learning techniques may provide a useful means for isolating the predictive measurements.

In this paper, we have considered two real world applications. Both are characterized by high-dimensional data. In each case, we were able to infer new knowledge that augments knowledge based on current human expertise.

In the first case study, the results of the rule induction experiments provided a good starting point for developing a metric for detecting a subset of the bad disk drives. This was made possible by the fact that the outputs of a rule induction system are parsimonious and interpretable. This permitted us to perform further analyses on the induced rule set to extract a promising sub-component. The fact that such analyses are possible make rule induction methods an excellent choice when explanations of decisions are critical.

The results of the disk drive analysis allow for a subset of the bad disk drives to be detected with high likelihood, prior to the expensive RunIn test. The nature of the RunIn test and the fact that this is a high volume operation, suggest that even the application of this simple test may result in a sizable reduction of the total manufacturing costs, in absolute terms.

In the second case study, we considered a highly complex communications network, and analyzed its behavior over time. To evaluate network performance, we have developed measurements that are sampled for the complete network during regular time periods. The analysis did produce strong predictors. Our method can be very useful in the diagnosis of problems in networks and in improving the performance of networks.

Both case studies involved intensive computer processing of very large volumes of data. In both objectivity and pattern matching capability, such efforts are clearly beyond the capabilities of human processing and experience. With increasing volumes of data and computing capabilities, the application of compute-intensive learning models may have an increasingly strong impact on real world applications.

References

- [1] T. Anand and G. Kahn. SPOTLIGHT: A Data Explanation System. In *Proceedings of the Eighth IEEE CAIA*, pages 2–8, 1992.
- [2] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, Monterrey, Ca., 1984.
- [3] P. Clark and T. Niblett. The CN2 Induction Algorithm. *Machine Learning*, 3:261–283, 1989.
- [4] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.
- [5] R. Galen and S. Gambino. *Beyond Normality: The Predictive Value and Efficiency of Medical Diagnoses*. John Wiley & Sons, New York, 1975.
- [6] P.J. Hayes, P.M. Andersen, I.B. Nirenburg, and L.M. Schmandt. TCS: A Shell for Content-Based Text Categorization. In *Proceedings of the Sixth IEEE CAIA*, pages 320–326, 1990.
- [7] M. James. *Classification Algorithms*. John Wiley & Sons, 1985.
- [8] J.L. McClelland and D.E. Rumelhart. *Explorations in Parallel Distributed Processing*. The MIT Press, 1989.
- [9] R. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains. In *Proceedings of the AAAI-86*, pages 1041–1045, 1986.

- [10] G. Pagallo. Learning DNF by Decision Trees. In *Proceedings of the Eleventh IJCAI*, pages 639–644, 1989.
- [11] J.R. Quinlan. Generating Production Rules From Decision Trees. In *Proceedings of the Tenth IJCAI*, pages 304–307, 1987.
- [12] B.D. Ripley. Statistical Aspects of Neural Networks. In *Proceedings of SemStat (Séminaire Européen de Statistique)*. Chapman & Hall, 1992. To Appear.
- [13] R. Sasisekharan, Y.K. Hsu, and D. Simen. SCOUT: An Approach to Automate Diagnoses of Faults in Large Scale Networks. In *Proceedings of IEEE GLOBECOM '93*, 1993.
- [14] C. Stanfill and D. Waltz. Statistical Methods, Artificial Intelligence, and Information Retrieval. In P. Jacobs, editor, *Text-Based Intelligent Systems*. Lawrence Erlbaum, 1992.
- [15] S. Weiss and N. Indurkha. Reduced Complexity Rule Induction. In *Proceedings of the Twelfth IJCAI*, pages 678–684, 1991.
- [16] S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn*. Morgan Kaufmann, 1991.