Volume 08 Issue 04 Published, November 17, 2004 ISSN 1535-864X

Intel® Technology Journal

Toward The Proactive Enterprise

Advancements and Applications of Statistical Learning/Data Mining in Semiconductor Manufacturing

A compiled version of all papers from this issue of the Intel Technology Journal can be found at:
http://developer.intel.com/technology/itj/index.htm

Advancements and Applications of Statistical Learning/Data Mining in Semiconductor Manufacturing

Randall Goodwin, Technology and Manufacturing Group, Intel Corporation
Russell Miller, Technology and Manufacturing Group, Intel Corporation
Eugene Tuv, Technology and Manufacturing Group, Intel Corporation
Alexander Borisov, Sales and Marketing Group, Intel Corporation
Mani Janakiram, Technology and Manufacturing Group, Intel Corporation
Sigal Louchheim, Information Services and Technology Group, Intel Corporation

Index words: statistics, machine learning, data mining

ABSTRACT

Knowledge Discovery in Databases (KDD) is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data [12]. Sometimes referred to as Data Mining, Machine Learning, or Statistical Learning (which we will use in this paper), it has many diverse applications in semiconductor manufacturing.

Some of the challenging characteristics of semiconductor data include high dimensionality (millions of observations and tens of thousands of variables), mixtures of categorical and numeric data, non-randomly missing data, non-Gaussian and multimodal distributions, highly non-linear complex relationships, noise and outliers in both *x* and *y* dimensions, temporal dependencies, etc. These challenges are becoming particularly acute as the quantity of available data is growing dramatically. To address these challenges, statistical-learning techniques are applied.

We begin the paper with a description of the problem, followed by an overview of the statistical-learning techniques we use in our case studies. We then describe how the challenges presented by semiconductor data were addressed with original extensions to tree-based and kernel-based methods. Next, we review four case studies: house sales price predictions, throughput time prediction, signal identification/separation and unit-level speed prediction. Finally, we discuss how enterprise-wide statistical models form a foundation for intelligent, automated decision systems, and we describe applications currently under development within Intel Corporation.

INTRODUCTION

In 1965, Gordon Moore, one of the co-founders of Intel Corporation, observed an exponential growth in the number of transistors per integrated circuit and predicted the trend would continue. Moore's Law, as the relationship was named, has been maintained to the present time and is expected to continue through the rest of the decade. A plot of the relationship is shown in Figure 1 for Intel microprocessors.

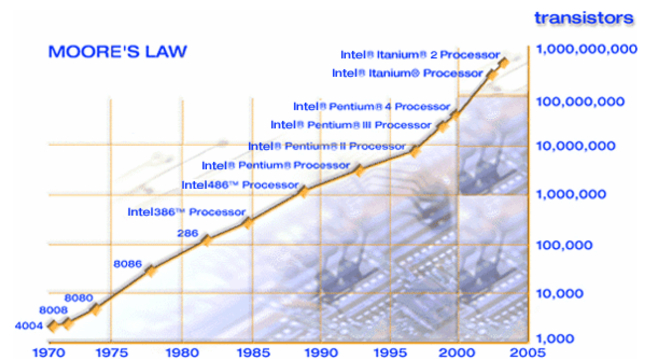

Figure 1: Moore's Law

To keep pace with Moore's Law, the semiconductor industry has relied upon many technical innovations and has grown significantly in complexity. The technological advances have been accompanied by an exponential growth in the quantity of data collected and stored during the manufacturing process. Examples of semiconductor manufacturing data include lot transactions and timestamps, process and equipment data, in line metrology data, electrical in-line test (e-test), wafer sort, and final electrical test/performance binning. Data may be at the lot level, wafer lever, or even the unit level. As Intel

Corporation ships several million microprocessors per quarter from its worldwide factories, the quantity of data stored in databases is now measured on the order of terabytes.

Traditional statistical approaches have served the industry well for many years and will continue to have an important place in semiconductor manufacturing. Often, a simple data plot, control chart, linear regression, or analysis of variance will tell an important story and enable the needed process discovery and controls. Design of Experiment (DOE) methods will also continue to play a key role in technology development and process health investigation/optimization. However, for multivariate/mixed data-type modeling of non-linear relationships, such as maximum functional frequency, lot throughput time and unit-level final bin classification, statistical-learning methods are required.

Our first case study is an illustrative one from outside the semiconductor domain on predicting the future sales prices of homes. When buying a home, consumers are guided by many factors such as location, square feet of the home, size of the lot, pool, etc. Real-estate agents and appraisers will readily provide “comps,” short for comparisons, to existing homes that have recently sold to help the buyer (and seller) determine the value of a home. Traditional multiple regression techniques can be used to model home prices; however, using an advanced statistical-learning method, lower errors are achieved, modeling times are reduced, and requirements for domain-specific knowledge and modeling techniques are eliminated.

Our second case study focuses on a key measure of Fab performance–cycle time (CT), sometimes referred to as throughput time (TPT). In semiconductor manufacturing groups of wafers are processed together and move through the various processing steps in a Fab lot. Given the implications of multiple lot reentrant steps, a high mix of different products, lots with different priority flags, equipment preventive maintenance schedules, set-up times, etc., the ability to understand and predict lot TPT would immensely benefit factory planning and scheduling. Traditional approaches for predicting the remaining TPT of a lot include static linear modeling using Little’s Law [13] and simulation techniques. However, static models do not handle stochastic variables, and simulations require large amounts of computing resources in order to predict the remaining cycle time. In this example, data-mining/statistical-learning methods use historical data to predict individual lot cycle-time by comparing key characteristics of a lot in progress to lots that have completed the target prediction operation. Results show that application of pre-clustering and a single decision tree results in better TPT predictions than with previous approaches.

The third case study focuses on utilizing statistical-learning technologies for signal identification/separation and for Advanced Process Control Systems (APCS). In this case study we are interested in identifying the key sources of variation of Sort Fmax–a measure of the maximum functional speed (frequency) of an individual microprocessor while still in wafer form. We are not only interested in identifying, ranking, and separating the sources of non-random variation, but also determining how the effects change over time. In our example, we use simulated data typical of those found in high-volume CPU manufacturing. Small Fmax “signals” were embedded in several simulated operations on a baseline normal variation (noise). By using advanced tree-based statistical-learning techniques and our newly developed extensions, we are able to separate out the subtle signals affecting Fmax not possible with traditional statistical-learning approaches or commercial data-mining software packages.

The final case study focuses on predicting the speed of individual microprocessors at the final test step. Natural process variation and both non-random and random defects result in microprocessors that are functional at different speeds. Predicting the final test outcome and speed bin at the unit level early in the flow presents unique challenges to traditional approaches; yet, such a prediction would yield clear benefits in planning, downstream test flow optimization, signal identification, yield improvement, advanced process control, and final assembly optimization strategies. A key enabler to this application has been the ability to trace individual units through key operations in the entire Fab and assembly test manufacturing flows. Unit-level numeric and categorical data (variables) form a basis for both training the models (when the final class test outcome is known) and predicting the final class test result of upstream units. Since the data are at the unit level, and hundreds of variables (or even thousands) are used for generating predictions, the dimensionality of the data sets is very large. Furthermore, the data often contain missing values from dynamic sampling schemes, outliers, temporal relationships, non-linear relationships and even dynamic sets of important variables. The novel extensions to tree- and kernel-based statistical-learning methods enable robust, accurate unit-level bin classification nearly impossible with traditional approaches or commercial data-mining software packages.

OVERVIEW OF STATISTICAL-LEARNING METHODS

In statistical- or machine-learning methods we are usually given an object with a set of variables/attributes, often

called “inputs” or “predictors” and a corresponding target, often called “response” or “output” values. The goal is to build a good model or predictive function capable of predicting the unknown, future target value, given input values.

When the response is numeric, the learning problem is called “regression.” When the response takes on a discrete set of k non-orderable categorical values, the learning problem is called “classification.” In predictive learning one uses data to build a good predictive model. A representative “training” data base with all response and predictor variables that have been jointly measured is assumed to exist. A “learning” procedure is applied to the training dataset in order to extract a good predicting function. There are many commonly used learning procedures including linear/logistic regression, neural networks, kernel methods, decision trees, etc. A technical overview of modern learning techniques is provided in [1].

Recently there has been a revolution in the field of statistical learning inspired by the introduction of three new approaches: the extension of kernel methods to support vector machines [4]; the development of reproducing kernel Hilbert space methods [5]; and the extensions of decision trees by application of boosting [6,7], bagging, and Random Forest (RF) techniques [2,3].

The complexity of the underlying data adds significant challenges when developing models for industrial applications. This includes mixed-type variables with blocks of non-randomly missing data, categorical predictors with a very large number of levels (hundreds or thousands). Very often datasets are extremely saturated: there are a small number of observations and a huge number of variables (tens of thousands). Both predictors and responses normally contain noise and mislabeled classes. Both regression and multi-level classification models are of interest. Thus, a universal, scalable and robust learner is needed.

Decision trees are one of the most popular universal methods in machine learning/data mining, and they are commonly used for data exploration and hypothesis generation. Classification and Regression Trees (CART), a commonly used decision-tree algorithm [8], use recursive partitioning to divide the domain of X variables into sets of rectangular regions. These regions are as homogeneous as possible with respect to the response variable and fit a simple model in each region, either majority vote for classification, or a constant value for regression [8]. The resulting model is a highly interpretable decision tree. Some of the principal limitations of CART are low accuracy, because piecewise, constant approximations are used, and high variance/instability.

One problem faced at Intel Corporation is multilevel classification problems (see the BinSplit case study below) with categorical predictors of high cardinality (m unordered values). Tree algorithms would need to evaluate $2^{(m-1)}$ possible partitions of the m values of the predictor into two groups. When m is large it becomes computationally intractable to evaluate all unique partitions. This problem is addressed in [9], and the algorithm is implemented in an internally developed set of statistical-learning algorithms. To reduce the high computational requirements, a hybrid clustering scheme (k-means and agglomerative) with a novel generalized distance metric was developed. This dynamic preprocessing method resulted in an efficient, computationally fast way to discover a small number of natural partitions of levels for such variables that have similar statistical properties in terms of categorical response.

Recent advances in tree-based methods such as Multiple Additive Regression Trees (MART) [7] and RF [13] have proven to be effective universal learning machines. Both are resistant to outliers in X-space, both have efficient mechanisms to handle missing data, both are competitive in accuracy with the best-known learning algorithms in regression and classification settings, and both handle mixed data types naturally. However, MART uses an exhaustive search on all input variables for every split and every tree in the ensemble, and it becomes extremely expensive computationally to handle very large numbers of predictors. At the same time, RF shows significant degradation in accuracy in the presence of many noise variables.

To address the computational limitations of MART and the susceptibility of RF to noise variables, a fast hybrid method using dynamic feature selection was proposed [10]. This method features a stage-wise stochastic boosting of shallow random trees built on a small intelligently sampled subset of variables. The method can be applied to noisy, massive regression and classification problems and has excellent predictive power. It is comparable to the best-known learning engines. Based on our experience, this combination of speed, accuracy, and applicability makes this hybrid method one of the best universal learners available. The hybrid method produced very competitive results at the 2003 Neural Information Processing Systems (NIPS) conference competition on feature selection in data sets with thousands of variables. There were over 1600 entries from some of the most prominent researchers in machine learning. Another method under development, based on an ensemble of kernel ridge classifiers [11], ranked as the second best entry in the NIPS 2003 competition, ahead of submissions from the world’s best research universities.

Sometimes, it is necessary to identify patterns in a given set of predictors and/or reduce the dimensionality of the predictor variables. In this case, there are no response variables and the technique often referred to as unsupervised learning or clustering is used to find and group data into similar sets of observations. A hierarchical method is used to form and identify appropriate numbers of clusters based on training data. The nearest neighbor method is used (k-means) for assigning test data to clusters formed. Our second case study evaluated clustering techniques in addition to CART for TPT prediction analysis.

Many of the statistical algorithms discussed throughout this paper and used in the case studies have been integrated into an internally developed Windows*-based statistical-learning platform optimized for Intel Architecture, called Interactive Data Exploration and Learning (IDEAL). The application is updated as new algorithms are developed, and it is validated using a variety of internal test data sets.

PREDICTING THE SALES PRICE OF HOMES

Introduction

In this case study we use the buying and selling of a home to illustrate statistical-learning technologies. Some typical questions asked of a real-estate agent when buying or selling a home are as follows. What is my home currently worth? What is the fair price of that new home I am considering purchasing? Are the sellers asking too much or is it a good buy? Instinctively, we know many factors (variables) influence the value of a home: the size of the home and lot, its location, the home features and upgrades, its proximity to schools, the age of the home, landscaping, pool, number of rooms, garages, etc. Even factors such as the real estate agent of the buyer and seller likely influence the final selling price of a home.

Approach

In this case study we used a small set of both numeric and categorical variables to "learn" and then predict the future sales price of homes in Tempe, Arizona. This is an example of regression, where the target variable (sometimes called the response or Y variable) is numeric. The case study was used to generate interest in machine-learning technologies and is currently used in an internally developed statistical-learning class.

* Other brands and names are the property of their respective owners.

The first, and often the most difficult step in developing any statistical model is extracting and preparing the data. In this case, data on recent home sales in the area of interest were available from public sources such as the local newspaper and the county assessor's office. Data on approximately 1,300 homes were collected, and the data included such variables as the square footage of the home and lot, the size of the pool (set to zero if the house had no pool), the year the house was built, the sales date, and the subdivision name. A sample of the data is shown in Figure 2 along with a simple bivarate plot of one of the variables vs. the sales price (Figure 3).

Table 1: Home sales data

Subdivision Name	Land Size	Livable sq. ft.	Pool	Built	Sold Date	Sale Price ($)
TEMPE ROYAL PALM	6,042	1,391	0	1984	10/7/2003	$177,000
TEMPE ROYAL PALM	8,246	2,406	450	1988	5/28/2003	$259,900
COVENTRY TEMPE	10,812	3,456	600	1998	3/27/2003	$470,000
WARNER RANCH VIL	3,742	1,435	0	1985	10/1/2002	$147,500
ESTATE LA COLINA I	16,069	3,239	750	1980	8/1/2002	$334,500
TUSCANY	12,785	3,342	500	1996	5/1/2002	$487,000
PECAN GROVE ESTA	7,427	2,570	0	1990	1/1/2002	$238,000
RAINTREE UNIT 2 LO	16,575	2,638	400	1982	4/1/2001	$380,000
CIRCLE G RANCHES	36,647	4,057	500	1980	4/1/2001	$585,000
ALTA MIRA 1 LOT 1-	9,509	2,475	480	1983	1/1/2001	$204,000
ESTATE LA COLINA I	12,637	2,683	425	1981	11/1/2000	$279,000

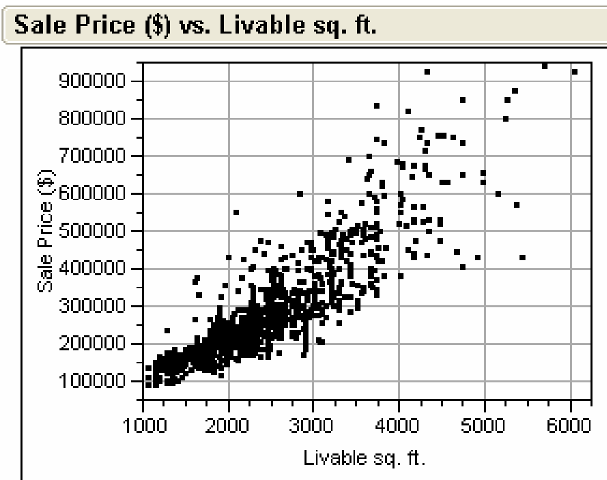

Figure 2: Sales price vs. livable square feet

To make the comparison to other statistical approaches fair and represent how the statistical model could be used in real life, the 1,300 home data set was manually divided into two parts based on the home sales date. The "training" data set included all but the 50 most recent home sales. The last 50 home sales were used as the test set to validate the model and quantify the error.

To compare with existing approaches, the training and test datasets (with the sales price withheld) were given to several statisticians within Intel Corporation to develop predictive models. Some statisticians used the data-mining

capabilities available in commercial software, while others applied traditional regression techniques. The traditional models took time to develop and required intermediate calculations such as average appreciation rates, assessment/screening of outliers, even driving through certain neighborhoods to view the homes in an attempt to build a more accurate model.

IDEAL was also used to create a gradient boosted tree (GBT) model to predict the sales price of homes in the validation data set. The mean error was compared between all modeling methods.

Results

A plot of the results achieved on the validation data set is shown in Figure 4. Using GBTs in IDEAL, the mean error was $14,473, while the error for other models ranged from $17,096 to $43,275. The time taken to model and produce the predictions using IDEAL was measured on the order of minutes, while most of the other statistical models took significant non-computational time to create and in some cases the statisticians had to manually intervene to identify and remove outlier observation(s) in the training data set.

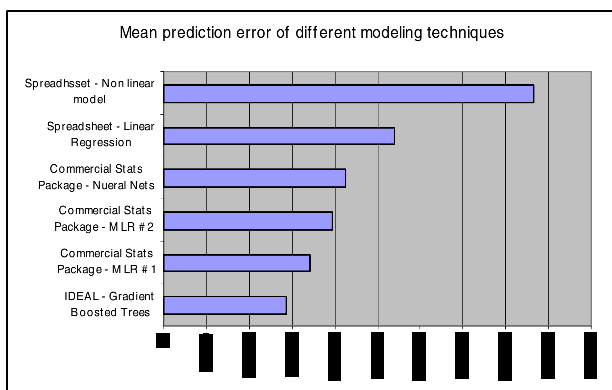

Figure 3: Prediction "contest" results

In addition to modeling speed and improved accuracy, IDEAL produced other outputs such as an explorable/drillable single decision tree (for single tree models), a variable importance pareto, and variable dependency plots. Examples of a variable importance pareto and dependency plot are shown in Figures 4 and 5. These tools aided in better understanding the relative variable importance and variable interactions/relationships present in the data.

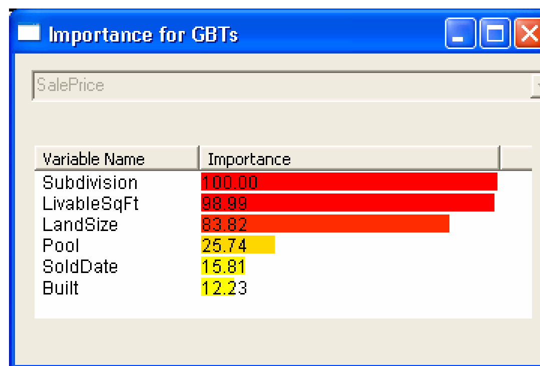

Figure 4: Variable Importance Pareto

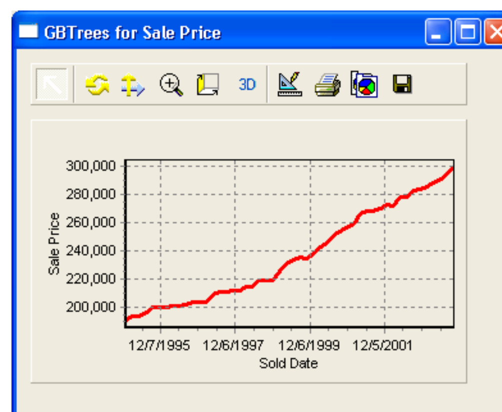

Figure 5: Dependency plot

Conclusion

Even in this simple example, where there are only six input variables, we can see the power of advanced machine-learning technologies. Because the data sets in semiconductor manufacturing are much more challenging, and the variable relationships often times non-linear, the use of statistical-learning techniques for accurate regression/classification and data exploration is a requirement.

THROUGHPUT TIME PREDICTION

Approach

A cross-functional team comprising groups from Intel and Arizona State University developed a proof of concept TPT data-mining technique using generic clustering and decision-tree techniques. Most of these techniques use historical data to provide prediction for current lots. This data-mining tool learns from past patterns in the factory and categorizes the current flow of products using its stochastic characteristics and existing data in order to predict lot TPT. To do this, the cumulative time of historical and existing lots by critical operation and route is used in addition to data on work in progress (WIP), and lot priority. The WIP values correspond to the number of

lots waiting to be processed at the critical operations at a given time. A typical data table in a manufacturing execution system database contains rows for each operation and each lot in production. Each row has information identifying the lot, the current operation, the production route, the time moved in to the current operation, the time moved out from the previous operation, and the time out from the current operation. There are also variables describing the type of production lot, the product type, and lot priority. There can be more than 400 rows for each lot with more than 15 columns of data, thus yielding more than 6000 variables to describe how an individual lot moves through the factory. A lot's cycle time for an operation can be described as queue time plus the tool processing time. Queue time is calculated as operation start minus previous operation out. Processing time is calculated as current operation out minus operation start. Queue time and process time were summed across all intermediate operations to get an aggregate cycle time between the critical operations.

$$Aggregate\ CT = \sum_{stepi}^{bottleneck} CT_i \qquad (1)$$

A lot velocity vector, which is a vector of aggregate cycle times for each critical step, was created, and different WIP measures were recalculated relative to move-in and move-out of the critical operations. Finally, the rows of data describing the individual steps for each lot were combined into a single row. Lots that did not have complete information were eliminated along with test or engineering lots. However, all filtered lots were included in the WIP counts since their presence affects the cycle time of production lots. Figure 6 pictorially represents the data-mining techniques and approaches adopted for lot TPT prediction.

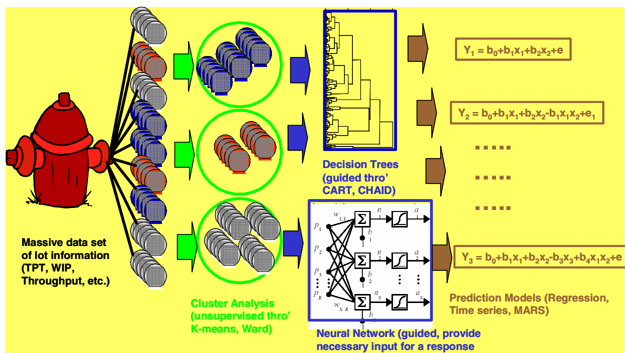

Figure 6: Data-mining techniques applied to lot TPT prediction

Little's Law states that on average CT = WIP/TP, where CT is the Cycle Time and TP is the throughput (processing rate) of an operation. Often the precision and reliability of the machines enable one to assume throughput to be nearly constant for a given machine. With this simplifying assumption, the cycle times of lots processed by the same tool are proportional to the ratio of the WIP as shown in equation 2.

$$CT_b = (WIP_b / WIP_a) * CT_a \qquad (2)$$

In a linear production process with a first-in, first-out (FIFO) scheduling rule, WIP remains constant for each lot at each tool, and the cycle time of a lot for the series of tools is proportional to the ratio of WIP. The accuracy of this prediction depends on the variability in machine throughput. However, many production processes allow tools to be used in multiple operations, in which case WIP for a lot is not constant at every tool. In reentrant cases a lot may leave a tool at one step and return to the same tool several steps later. The WIP for the second pass through the tool includes new lots plus the lots making their first pass through the tool.

Although equation 2 cannot be used directly for cycle-time prediction, the measures of cycle time and WIP should be interchangeable as variables due to their proportional relationship. Also, lots that encounter similar values of WIP should have similar cycle times, and these can be predicted by comparing a new lot's cycle time at critical steps in the process to the cycle times of lots that have completed the process. The average cycle times for similar lots would predict the cycle time for the target lot. From the transactional data it is easier to derive an estimate of WIP state. To that end, averages of cycle times for adjacent steps, within a window of time, provide information about throughput and WIP. For the steps just ahead of the target lot, the average cycle time actually provides information about both the WIP ahead of the target lot and a measure of the throughput and WIP for prior lots. If the window is constructed to include lots that finish the current operation before the target lot and do not leave the operation before the target lot enters, then n counts all such lots, and n will be equal to the WIP for the target lot. Hence, we get the following equation:

$$Avg\ CT = \frac{\sum_{i=1}^{n} CT_i}{WIP} \qquad (3)$$

Given a target lot at step i, we can calculate the average cycle time at step i for neighboring lots. Each intermediate cycle time has an associated variance. The variance then for predicting the cycle time from the ith tool to process completion is as follows:

$$Var(\sum_{i=j}^{m} CT_i) = \sum_{i=j}^{m} Var(CT_i) + \sum_{i=j}^{m} \sum_{h=j}^{m} Cov(CT_i, CT_h)\ i \neq h. \qquad (4)$$

In the data considered here, the covariance between intermediate cycle times was positive. Hence, the variance of the sum of intermediate cycle times increases as the number of intermediate steps increase. As expected, predictions of TPT made nearer to the end of the process have lower error rates since the amount of variability has decreased.

Results

In general, we found that data could be partitioned in a way that provides good predictions for future observations. However, the accuracy of the prediction varied from method to method. If cluster assignment is based on Euclidean distance from cluster center, prediction of multiple steps can be made from any current step using only one set of clusters. K-Nearest-Neighbors Prediction of multiple steps can be made from any current step by using same nearest neighbors. Regression trees perform automated handling of optimal tree size, and the tree must be constructed for each step to be predicted. These data-mining techniques were compared based on the mean absolute error and median absolute error (median is robust to outliers). CART and CART in Cluster performed better and provided prediction variability within two days as shown in Table 2.

Table 2: Comparison of model results

Method	Data Split	Median Absolute Error	Mean Absolute Error
KNN (K=5)	Random	2.10 days	4.88 days
KNN (K=10)	Random	2.35 days	4.91 days
CART	**CV**	**1.25 days**	**2.66 days**
Cluster (N=5)	Random	2.93 days	3.66 days
Cluster (N=10)	Random	2.60 days	3.38 days
CART in Cluster	**CV**	**1.09 days**	**1.65 days**
Neural Network	Random	2.55 days	3.61 days

The model validation also showed favorable results as shown in Figure 7 in which actual TPT is compared to predicted TPT. A demo product has been developed that combines these steps into a seamless process using a simple user interface.

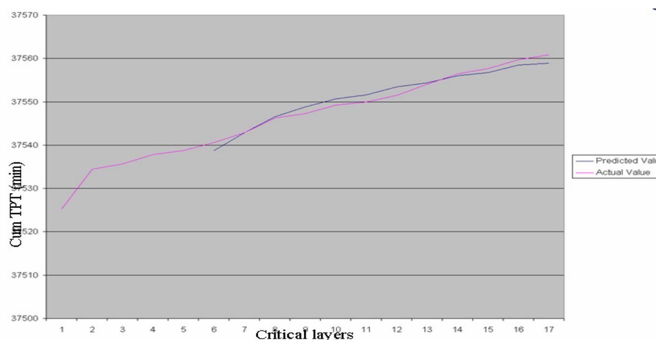

Figure 7: Comparison of actual TPT with predicted TPT

Conclusions

Analysis of data mining for lot TPT prediction on Fab data showed lot TPT prediction within two days and compared favorably with other static empirical models. Good predictions for lots currently in production can be obtained from similar lots that have already completed production. The team is currently evaluating a detailed pilot at one Fab. The powerful capability of this data-mining technique (Cluster and CART) is attracting many other internal customers. Some possible future projects include product health indicator, analysis/forecasting, demand planning, and process control.

SIGNAL IDENTIFICATION/SEPARATION

Introduction

Semiconductor fabrication is becoming increasingly complex, with routes stretching to several hundred process steps. Even with highly advanced process control systems in place, there is inevitable variation in yield and performance between and within manufacturing lots. A common practice is to associate this variation with equipment differences by performing analysis of variance at every process step where multiple pieces of equipment are used. Looking for operations where there are significant differences between process tools can reveal the sources of process variation.

Challenges

The one-at-a-time approach to equipment commonality studies has many shortcomings:

1. Most target variables of interest are affected by multiple process steps. For example, yield can be reduced by high particle counts at nearly any manufacturing step. The maximum frequency (Fmax) at which a part can run can be affected by a variety of lithography, etch, implant, and diffusion operations.

2. Lots are not distributed randomly across process tools. Often, material from one tool at one operation is run preferentially (or even exclusively) on a particular tool at another operation. In addition, some operations (notably lithography) are prone to running large blocks of the same product in a short time period, then not running any more for several weeks. Lots may also cycle through the same tool several times at different process operations.

3. The difference between tools can rarely be described by a constant offset. For example, lots from one tool are rarely a consistent 100 MHz faster than those from another tool over a significant period of time. Usually, the difference between process tools varies over time, sometimes dramatically. There is usually a

mixture of short-term and long-term trends in the data, often associated with the cycles for preventative maintenance.

Results

By using machine-learning techniques, we have overcome the above challenges, and can now look at all processing operations simultaneously, while also accounting for temporal trends in the data. The methodology we used is simple. At each process operation, we create two columns of variables: a categorical column indicating which process tool the lot was processed on, and a numeric column giving the time and date at which the lot was processed. These columns, which can number in the thousands, are fed into the learning engine along with the target variables of interest. We have found that stochastic, GBT models can give astonishingly accurate reconstructions of the time trends, even when there are significant differences at multiple operations.

These techniques are best demonstrated using simulated data, since the underlying trends in actual process data are unknowable. In this example, we simulated a hundred operation processes where five operations had significant differences between tools for Fmax. The programmed differences included steady-state offsets, saw-tooth patterns of short duration, step functions, and gradual drifts. Each lot had the same baseline frequency, to which the offsets from all of the affected tools were added. Additional random noise was also added.

One of the operations with an implanted signal is shown in the graph in Figure 8. Two of the five tools (entities) have square wave shape signals, while the remaining three tools have a consistent, zero offset. Other operations had different shapes of embedded signals.

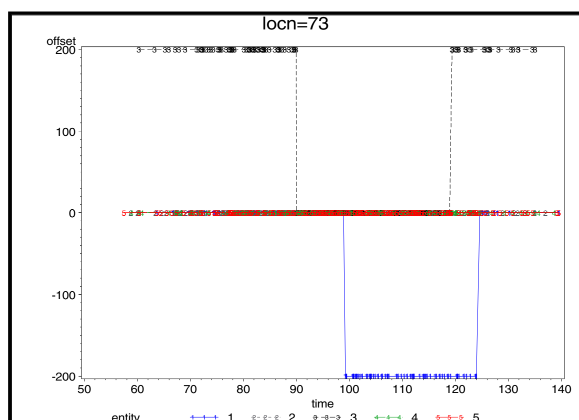

Figure 8: Implanted signals

Detecting which tools had non-random effects was the first challenge. Using stochastic, GBT models in IDEAL we were able to both identify which operations had embedded signals and accurately recover the patterns. To identify the tools and times with non-random patterns, we used the Variable Importance Pareto, which showed the relative variance reduction for all variables in the model. Variables with high variance reduction appeared highest on the pareto as in Figure 9.

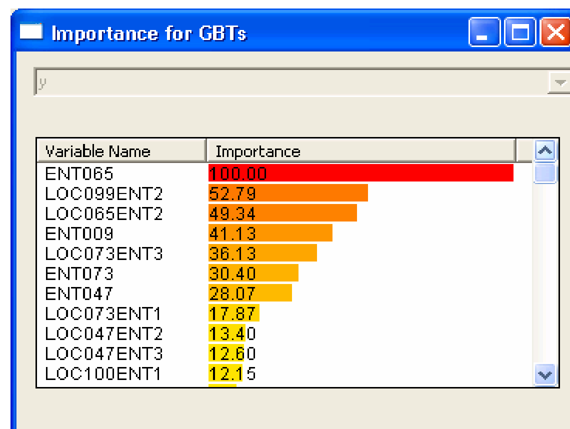

Figure 9: Variable importance

Signal separation is the next step. Figure 10 shows how the signal programmed in the data for one operation was lost in the added noise and the other valid embedded signals at other operations. Plotting just the response variable as a function of the time through the operation of interest did not show the important shift. Even using traditional approaches to average out the noise yielded only marginally better results–and required the upfront knowledge of which operations had signals.

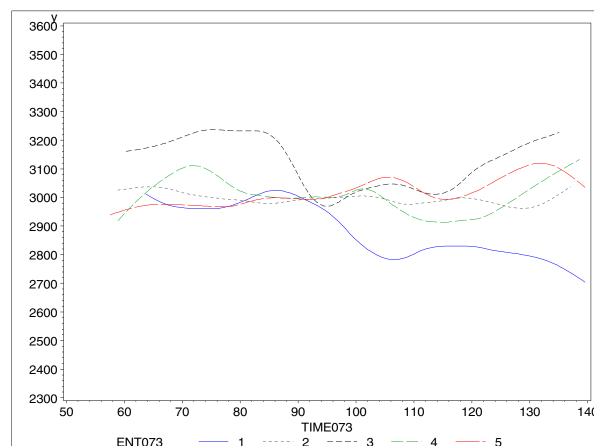

Figure 10: Hidden signal

Figure 11 depicts the recovered signal using GBT models in IDEAL. The recovery is not perfect due to the injection of random noise and the presence of valid signals from other operations. The recovery was far superior, however, to that given by a wide variety of classical statistical techniques, including multiple spline-based models.

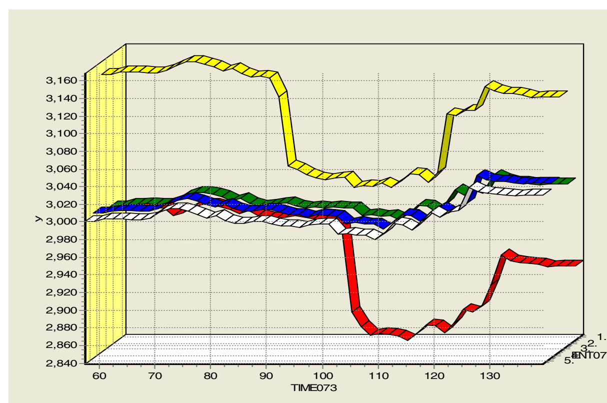

Figure 11: Recovered signals

The GBT-based machine-learning techniques are particularly attractive since they allow for both continuous and categorical target variables, and they allow for easy introduction of other predictor variables of mixed data types such as chemical vendors, queue times, or the results of inline metrology measurements.

UNIT-LEVEL BIN SPEED PREDICTION

Introduction

In the last case study we tackled a challenging application of statistical learning: accurately predicting the final test outcome (bin speed and yield) of individual microprocessors using upstream data from Fab and Sort. The results presented in this paper are for a stepping of the Intel® Pentium® 4 desktop processor on 130nm technology that is no longer in high-volume production. Analysis using the same techniques, with many additional variables and observations on current microprocessor generations has yielded similar, and in some cases better, results.

In this case, challenges emerge from a variety of factors such as the large number of observations and variables, non-randomly missing data (i.e., sampled), both categorical and numeric variables, presence of outliers, dynamic variable names, frequent process changes and improvements, non-linear variable relationships, etc. Even extracting the variables from databases and properly associating them with the appropriate unit level presented many challenges.

One traditional approach to classify the final speed of a CPU is to fit multiple linear logistic regressions by using a limited quantity of upstream numeric predictor variables such as Fmax and leakage current, measured at the wafer sort operation. However, these models assume prior knowledge of the underlying distributions, do not effectively address categorical variables, and require periodic, manual recalculations. Recent voltage and power optimization test strategies also invalidate many distribution assumptions.

Approach

To predict the final unit-level outcome, we first needed to extract and prepare the unit-level data set that contained both the predictors and responses (X's and Y's) for training the models. Several internally developed tools exist for extracting and joining unit-level data from databases across multiple factory data sources. These tools have improved in speed and capability as unit-level traceability has become the norm for semiconductor manufacturing. Even more specialized data marts have been created to lower the accessibility hurdles and improve speed and scalability.

We defined our response variable as "final speed if pass or fail" which resulted in six unique classes of units. The distribution of the response variable in the training dataset is shown in Figure 12.

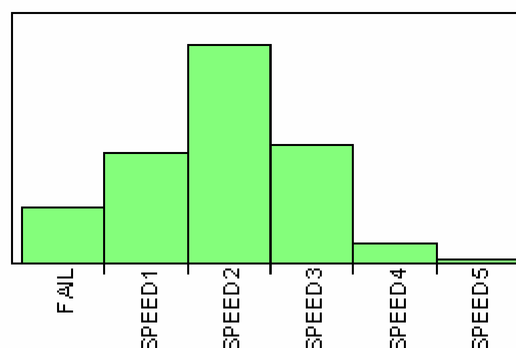

Figure 12: Distribution of response variable

For this multilevel classification problem we used GBT models in IDEAL to train the model. The model error was assessed using cross validation, i.e., using the model to classify observations that were withheld from the data set during the training process. Cross validation is a key component of IDEAL.

A bivariate plot of two of the continuous X variables (in arbitrary units) vs. the response variable (color/symbol of the points) is depicted in Figure 13. It is meant as a visual aid. If we had been using only two numeric variables to classify the units into the six possible outcomes, we would essentially be determining the set of two-dimensional

® Intel Pentium 4 is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

classification boundaries that minimize error on the test data set.

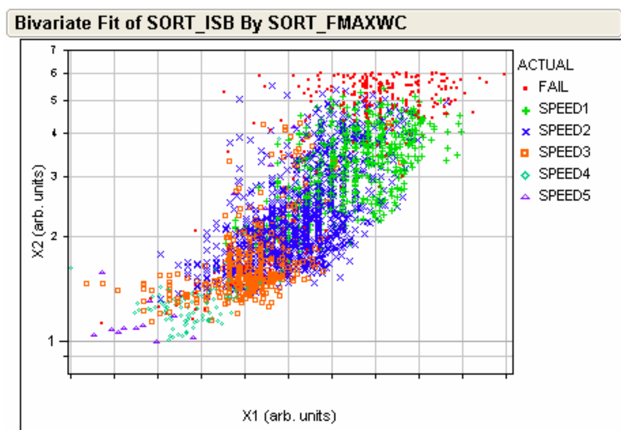

Figure 13: X2 vs. X1 (arb. units)

However, we are using many continuous variables as well as categorical variables as predictors, so the two dimensional classification boundaries now become an *n* dimensional classification hyper surface, where *n* is the number of variables in the model.

The overall error for a multilevel classification is determined by the misclassification percent across all classes of the target variable. In other words, it is the percent of total observations in the validation data set that were misclassified by the model. The goal of any classifier algorithm is to minimize misclassification. For multilevel and unbalanced classification problems, the rare classes may be “sacrificed” to minimize overall misclassification. IDEAL provides options in GBT models to change the relative weight, sometimes called misclassification penalty, of rare classes as the model is built. Overall misclassification may increase slightly; however, misclassification on rare classes is reduced. This is often the case when we are modeling rare classes such as failing units or a microprocessor speed bin that is faster than the one currently on the market. Other GBT options are available in IDEAL and were used for fine tuning the classification model. These included the learning rate, tree depth, number of iterations, and dynamic feature selection [10].

Results

The results from the case study presented in this paper show that low unit-level misclassification rates are achievable using numeric and categorical predictors from upstream (Fab and Sort). The overall cross-validated misclassification rate was 24% for the test data set. The misclassification matrix is shown in Table 3. The value in each cell is the percent of total observations in a given class that were predicted for each of the possible classes.

Table 3: Misclassification matrix

		PREDICT					
	Row %	FAIL	SPEED1	SPEED2	SPEED3	SPEED4	SPEED5
ACTUAL	FAIL	68.13	10.01	14.21	6.20	1.20	0.25
	SPEED1	2.26	82.44	15.30	0.01	0.00	0.00
	SPEED2	0.45	8.44	84.75	6.34	0.01	0.01
	SPEED3	0.09	0.73	20.07	77.46	1.62	0.03
	SPEED4	0.25	0.76	5.89	22.40	69.44	1.26
	SPEED5	0.88	7.51	27.69	11.05	13.25	39.62

For this data set, even a significant portion of true fails was predictable based on upstream Fab and Sort data. The decision to build die with higher probability of fail into units was taken because of an analysis of the relative die and final unit costs and other business considerations. Optimization of the decision process using financial variables and other business rules/constraints is currently an active area of research and development.

Although the primary goal of this case study was to determine how accurate a classification model can be created using statistical-learning techniques, several other benefits were also realized during the machine-learning process. A single tree model, although much less accurate than GBT models, was created in seconds using IDEAL. The single tree enabled visualization of the model and offered insights into non-obvious variable relationships. Also, IDEAL calculated normalized variance reduction, so we are able to see which variables are the most important in predicting the final speed. Lastly, the dependency plots are able to show the effect of an individual variable after averaging out the effects of all the other variables. Although the results are not presented here, these outputs from IDEAL were useful in identifying sources of equipment variation that impacted the final classification result–very similar to the previous case study on signal identification/separation.

Using more recent data on different microprocessor lines misclassification rates have been reduced from 20% down to 10%.

DISCUSSION

The results of the case studies demonstrate how data-mining/statistical-learning methodologies are being used at Intel Corporation to convert large amounts of semiconductor manufacturing data into real, actionable knowledge. These case studies only scratch the surface of the possible applications of these methodologies, many of which are active areas of research and development. Examples include simultaneous prediction of remaining cycle time and final test results, factory WIP prioritization and optimization, reformulation of yield and speed prediction models into unit-level predicted profit, autonomous learning prediction and optimization, and advanced multivariate process control systems.

All the applications mentioned above require robust and efficient statistical-learning technologies that can address the challenges of semiconductor data. IDEAL, developed internally with these requirements in mind, has demonstrated leading-edge capabilities in prediction accuracy, modeling speed, and model interpretability. Indeed, internal benchmarks have shown IDEAL to be faster and more capable when compared to commercial statistical, data-mining packages.

CONCLUSION

Utilization of data-mining and advanced statistical-learning methods in the semiconductor industry will continue to grow and will play an important role in maintaining Moore's Law. With each successive process generation, semiconductor manufacturing becomes more technologically complex. Concurrently, the quantity, complexity, and availability of data also march forward. These advanced methods provide us the means to understand and extract key lessons from our complex data.

ACKNOWLEDGMENTS

The authors gratefully acknowledge IT, R&D, and analysis help from Arizona State University, particularly Dr. George Runger and his students.

REFERENCES

[1] Hastie T., Tibshrirani R., and Friedman J., *The Elements of Statistical Learning*, Springer, New York, 2001.

[2] Breiman, L., "Bagging predictors," *Machine Learning 26*, 123-140, 1996.

[3] Breiman, L., "Random forests, random features," *Technical Report*, University of California, Berkeley, 2001.

[4] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[5] Wahba, G., "Spline Models for Observational Data," *SIAM*, Philadelphia, 1990.

[6] Y. Freund and R. E. Schapire, "*Experiments with a New Boosting Algorithm*," in *Proceedings of International Conference on Machine Learning*, pp. 148-156, 1996.

[7] Friedman, J. H., 2001 "Greedy function approximation: a gradient boosting machine," *Annals of Statistics 29*, pp. 1189-1232, 1996.

[8] L. Breiman, J.H. Friedman, R. A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth Inc., Belmont, California, 1984.

[9] Tuv, E., Runger, G., "Pre-Processing of High-Dimensional Categorical Predictors in Classification Settings," *Applied Artificial Intelligence 17(5-6)*: 419-429, 2003.

[10] Borisov, A., Eruhimov, V., and Tuv, E., "Flexible Ensemble Learning with Dynamic Soft Feature Selection," forthcoming chapter in *Feature Extraction, Foundations and Applications*, editors: I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, Springer, New York, 2004.

[11] Torkkola, K. and Tuv, E., 2004 "Ensembles of Regularized Least Squares Classifiers for High Dimensional Problems," forthcoming chapter in *Feature Extraction, Foundations and Applications*, Editors: I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, Springer, New York, 2004.

[12] Fayyad, U., Piatetsky-Shapiro, G. and Padhraic Smyth, "From Data Mining to Knowledge Discovery: An Overview," Chapter 1 in *Advances in Knowledge Discovery and Data Mining*, pages 1–34, AAAI Press, 1996.

[13] Hopp, W. J and Spearman, M.L., "Factory Physics–Foundations of Manufacturing Management," The McGraw-Hill Companies, Inc., 1996.

AUTHORS' BIOGRAPHIES

Randall Goodwin graduated from Cornell University in 1992 with a B.S. degree in Applied and Engineering Physics. He joined Intel in 1992 and currently works in product and test technology development. One of his many interests is the application and optimal utilization of machine-learning technologies in semiconductor manufacturing. His e-mail is randall.s.goodwin at intel.com.

Russ Miller holds a B.A. degree in Physics from the University of Chicago, an M.S. degree in Statistics from Texas A&M, and an M.S.E.E. degree from Columbia University. He joined Intel in 1992 and has held a variety of positions in statistics, yield engineering, and strategic forecasting. His primary interest is improving the yield, reliability, and performance of high-volume microprocessors through the analysis of very large data sets. His e-mail is russell.miller at intel.com.

Eugene Tuv is a staff research scientist in the Enabling Technologies and Solutions Department at Intel. His research interests include supervised and unsupervised non-parametric learning with massive heterogeneous data. He holds postgraduate degrees in Mathematics and Applied Statistics. His e-mail is eugene.tuv at intel.com.

Mani Janakiram is a manager in the Enabling Technologies and Solutions Department at Intel. He has

18+ years of experience and has published 20+ papers in the area of statistical modeling, capacity modeling, data mining, factory operations research, and process control. He has a Ph.D. degree in Industrial Engineering from Arizona State University. His e-mail is mani.janakiram at intel.com.

Sigal Louchheim leads the Data Fusion research focus area in the Information Services and Technology Group. Her main research interests are Interestingness (what is interesting) in Knowledge Discovery and Data Mining. Sigal received her Ph.D. degree in Computer Science in 2003, her M.Sc. degree in Computer Science in 1996, and her B.Sc. degree in Mathematics and Computer Science in 1990. Her e-mail is sigal.louchheim at intel.com.

Alexander Evgenyevich Borisov was born in Nizhny Novgorod, Russia and received his Master's degree in mathematics (Lie algebras) at Lobachevsky Nizhny Novgorod State University where he is currently working on a Ph.D. degree in the area of context-free grammars. He currently works at Intel (Nizhny Novgorod) as a software engineer and researcher. His technical interests include artificial intelligence and data mining, especially tree-based classifiers. His e-mail is alexander.borisov at intel.com.

Copyright © Intel Corporation 2004. This publication was downloaded from http://developer.intel.com/.

Legal notices at
http://www.intel.com/sites/corporate/tradmarx.htm.

For further information visit:

developer.intel.com/technology/itj/index.htm

Copyright © 2004 Intel Corporation. All rights reserved.
Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.
For a complete listing of trademark information visit: www.intel.com/sites/corporate/tradmarx.htm