J Intell Manuf
DOI 10.1007/s10845-008-0148-7

Optimizing a batch manufacturing process through interpretable data mining models

Mark Last · Guy Danon · Sholomo Biderman · Eli Miron

Received: 12 December 2007 / Accepted: 23 June 2008

Abstract In this paper, we present a data mining based methodology for optimizing the outcome of a batch manufacturing process. Predictive data mining techniques are applied to a multi-year set of manufacturing data with the purpose of reducing the variation of a crystal manufacturing process, which suffers from frequent fluctuations of the average outgoing yield. Our study is focused on specific defects that are the most common causes for scraping a manufactured crystal. A set of probabilistic rules explaining the likelihood of each defect as a function of interaction between the controllable variables are induced using the single-target and the multi-target Information Network algorithms. The rules clearly define the worst and the best conditions for the manufacturing process, also providing a complete explanation of all major fluctuations in the outgoing quality observed over the recent years. In addition, we show that an early detection of nearly the same predictive model was possible almost two years before the end of the data collection period, which could save many of the flawed crystals. The paper provides a detailed description of the optimization process, including the decisions taken at various stages and their outcomes. Conclusions applicable to similar engineering tasks are also outlined.

M. Last (✉) · G. Danon
Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
e-mail: mlast@bgu.ac.il

G. Danon
e-mail: guyda@bgu.ac.il

S. Biderman
Rotem Industries Ltd., Dimona, Israel
e-mail: sbider@netvision.net.il

E. Miron
Nuclear Research Center–Negev, P.O. Box 9001, Beer-Sheva, Israel
e-mail: emiron@nrcn.org.il

Keywords Data mining · Process optimization · Rule induction · Predictive modeling · Information networks

Introduction

Outgoing quality assurance is a critical issue in irreversible manufacturing processes, where certain defects found in the final products cannot be removed or repaired. Consequently, each finished product that does not meet the quality requirements has to be disposed, or "scrapped", thus decreasing the overall "yield" of the manufacturing process. Though the processing of each manufacturing unit (e.g., a lot or a batch) involves exactly the same sequence of steps and each step is performed under a rigorously defined set of conditions, the final product quality may still be subject to significant variations due to a large number of uncontrollable factors such as changes in environmental variables (temperature, humidity, etc.). The complex, nonlinear nature of these processes often results in a poor understanding of the true relationships between controllable and uncontrollable factors on one hand, and the product quality parameters on the other hand.

In this paper, we present a novel process optimization methodology based on the Data Mining approach. The operating model relating the process quality (output variables) to controllable (manipulated) variables is constructed from past operational data using single-target and multi-target Information Network (IN) algorithms for induction of oblivious decision-trees (Maimon and Last 2000; Last and Maimon 2004; Last 2004). The proposed optimization methodology is applied to the data of a batch manufacturing process, which suffers from high variability of the outgoing yield.

The IN algorithms were chosen for predicting the process outcomes for several reasons. First, as shown in (Last and Maimon 2004), they tend to produce considerably smaller

models than other decision-tree algorithms of similar accuracy. The interpretability of the induced models is critical here for two main reasons: on one hand, the operators need a small set of simple rules to control the process parameters in the future; on the other hand, the process engineers are highly interested to explain the ups and downs in the process quality that were recorded in the past. Second, an Information Network associates *probability estimates* rather than categorical predictions with each leaf node. The leaf nodes with the highest probability of a successful outcome can be directly converted into sets of operational recommendations without any manual involvement of domain experts. An additional benefit of the Information Network methodology is the automated removal of irrelevant and redundant attributes from the induced models.

The rest of this paper is organized as follows. "Literature survey" section covers two related areas: optimization of manufacturing processes and induction of predictive data mining models from batch manufacturing data. "Crystals manufacturing process" section provides a brief description of the crystal manufacturing process, which served as our case study. "Optimization with Information Network models" section presents an overview of Information Networks and their suggested use for optimization of complex, nonlinear processes. Then, in "mining crystal quality data" section, we utilize the Information Network algorithms to induce models predicting the crystals quality as a function of controllable parameters. In "optimizing the crystal quality with IN models" section, we proceed with converting the predictive models into process recommendations, while showing how the same models can be used to explain the past behavior of the outgoing yield. Conclusions are outlined in the "Conclusions" section.

Literature survey

In the absence of an accurate mathematical model of the manufacturing process, it takes time to reverse each temporary decline in the outgoing quality. The classical statistical approach to the problem of optimizing the outcome ("response") of complex manufacturing processes, especially in the chemical industry, is the *Response Surface Methodology*, or *RSM* (Myers and Montgomery 2002). To optimize the response of a given process, the process engineer needs to determine the optimal levels of "independent variables", which are subject to his/her control. The Response Surface Methodology suggests an efficient Design of Experiments (DOE) strategy for exploring the space of process conditions, developing an approximate relationship ("response function") between the quality and process variables, and finally optimizing the levels of the process variables to meet the quality requirements. The RSM is a multi-phase process that builds mostly upon the results of carefully designed and potentially costly experiments rather than on the actual process measurements taken in the past.

Lin and Lin (2005) present a fuzzy logic based approach for optimization of a machining process with multiple responses. The entire machining parameter space is studied using a nine row orthogonal array. Multiple quality characteristics of the corresponding nine experiments are combined by a fuzzy logic unit resulting in "grey-fuzzy reasoning grades" of each configuration setting. The highest reasoning grade leads to the choice of the optimal machining parameters for the given process.

Liau et al. (2004) build an expert system for finding the optimal operating conditions of a crude oil distillation process. The operating model relating the product quality (output variables) to uncontrollable and controllable (manipulated) variables is constructed by applying the ANN (Artificial Neural Network) algorithm to collected experimental data. Though the induced operating model cannot be directly interpreted by the process engineers, it can be utilized by an optimization toolbox to solve the non-linear constrained optimization problem of the given process.

Data mining techniques (see Han and Kamber 2006) provide a valid alternative for control and optimization of manufacturing processes by utilizing the historical operational data rather than spending a considerable effort on expensive engineering experiments. Thus, Babu and Frieda (1993) use artificial neural networks (ANN) to induce predictive models from past operational data of an autoclave curing process used to manufacture composite materials. The variables associated with the process are partitioned into the following categories: the initial state of the system, input disturbances, manipulated (controllable) inputs, and intermediate measurements. The induced ANN model is used as a "black-box" predicting the expected quality of each batch. The batch neurocontroller is calculating the optimal values of the manipulated inputs using the predictive model and a standard gradient optimization package. The optimal process conditions for a given batch can be determined either off-line (before the start of the process) or on-line (using intermediate measurements from the batch). All optimization results in (Babu and Frieda 1993) were obtained using a simulation model of the actual process, since, as indicated by the authors, it is difficult to generate a good distribution of operational data for training the algorithm.

Another attempt to optimize parameters of a manufacturing process using an artificial neural network is done in (Cook et al. 2000), where the trained ANN model is again used as a quality predictor. However, in this work, the optimal values of the process parameters are determined by a genetic algorithm (GA). The combination of a neural network with a genetic algorithm is used in the different stages of a particleboard manufacturing process to obtain the desired strength

of the final board. Since a neural network is a "black-box" model, it cannot significantly improve the understanding of the actual relationships between process and product parameters. Not surprisingly, the ability of the ANN-GA tool to explain its recommendations is also quite limited.

One of the most natural ways to obtain an interpretable, "white-box" model of a manufacturing process would be decision-tree induction. Thus, Famili (1994) has used Quinlan's ID3 algorithm (Quinlan 1986) to induce decision-tree models of electrochemical machining (ECM) process. An upper and a lower limit were defined for each one of the 26 quality parameters resulting in 52 classification problems. The decision tree induced for each classification problem was converted into a set of production rules. However, the total number of obtained rules (320) was far beyond the capability of operators or process engineers to use. Thus, a significant amount of rules had to be manually removed or simplified based on a complex set of criteria. At the end of this process, only 10 useful rules (out of 320) were left and converted into recommendations for the process engineers and operators.

Hur et al. (2006) suggest a hybrid process diagnosis system, which infers cause-and-effect rules for the manufacturing process condition by using the C4.5 decision tree algorithm (Quinlan 1993) combined with a genetic algorithm (GA). The real training data was obtained from a coil-spring manufacturing company in Korea. If the current condition of the process is diagnosed as an abnormal condition, the most effective maintenance action is recommended by the system. To select an appropriate maintenance action, the authors construct a decision network which represents maintenance actions as possible paths from the detected abnormal node (abnormal condition) to the normal node (normal condition) based on the cause-and-effect rules inferred by hybrid learning. The most appropriate maintenance action is selected by the Analytical Hierarchy Process (AHP).

Last and Kandel (2001) apply the Information Network (IN) algorithm to the WIP (Work-in-Process) datacollected in a semiconductor plant. The induced models are aimed at predicting the line yield and the flow time of each manufacturing batch. The IN predictive accuracy is shown to be comparable with the C4.5 decision-tree algorithm, while C4.5 tends to produce much more complex models than IN. Fuzzy-based techniques of automated perception are further used for making the sets of if-then rules extracted from the IN models more compact and interpretable. Braha et al. (2007) show that a combination of classifiers, such as C4.5 and IN, trained on the same semiconductor dataset can increase the utility of a decision-making process related to each individual batch. In (Braha et al. 2007), two available actions include 'scrap a batch' vs. 'continue production', whereas the operating conditions of the batch manufacturing process remain changed.

The process optimization methodology presented in this paper has the following original contributions vs. the existing work:

- Using the compact and interpretable Information Network (IN) classification models for optimizing the quality of a batch manufacturing process. Previous works either used the "black-box" Artificial Neural Network (ANN) models (Babu and Frieda 1993; Liau et al. 2004; Cook et al. 2000) or the standard decision-tree algorithms (Famili 1994; Hur et al. 2006), which tend to produce incomprehensibly large sets of prediction rules.
- Introducing an automated methodology for optimizing a set of independent quality dimensions using a multi-target classification model. Previous works aimed at optimizing multiple quality parameters (such as Famili 1994) assumed a manual integration of several single-target classification models.
- Using the induced data mining models as an explorative tool for explaining the variability in the past failure rates. Such analysis, unmentioned in previous works on process optimization with data mining, can significantly contribute to the credibility of model recommendations in the view of process engineers.
- Finally, to the best of our knowledge, this study is the first published application of data mining techniques in the crystal manufacturing industry.

Crystals manufacturing process

The crystal growth division at Rotem Industries LTD has developed a unique technique (Gradient Solidification Method—GSM) for growing large sapphire dome-shaped crystals (see Horowitz et al. 1987; Biderman et al. 1991; Horowitz et al. 1993). This process significantly shortens the production time of Sapphire domes, which constitute the transparent "nose" of smart anti-aircraft missiles that rely on detection of Infra Red radiation.

The near-net shaped single-crystal Sapphire domes are grown in double-walled molybdenum crucibles. The crucible, loaded with sapphire powder and crackles, is placed in a vacuum furnace, which includes graphite heating element and molybdenum heat shields. The growth is performed by melting of the raw material via heating the crucible, followed by temperature reduction to obtain the various growth rates required at different stages of the process. Typical duration of growing a medium sized crystal (100 mm diameter and 50–70 mm height) is about one week.

The temperature gradients, which are required for proper growth, may cause quality problems since the upper part of the crucible can be exposed to elevated temperatures that enhance chemical reactions between the furnace

constituents. The reaction products could then enter the crystal, usually causing defects. The growth parameters are intentionally tweaked in an attempt to increase the yield. However, the interpretation (whether the parameter change improved the yield) is not immediate due to the long growth period and the non-deterministic occurrence of the problems.

In order to understand the influence of the various parameters, we built a database that includes full listing of the growth processes performed over recent ten years. About 50 variable parameters and 50 measured parameters are recorded for each growth. The rejects are related to one or more reason (of about 30 possible).

This database has been used for the data mining process.

Optimization with Information Network models

The single-target Information Network (IN)

The single-target Information Network (Maimon and Last 2000) is an oblivious tree-like classification model, which is designed to minimize the total number of input (predictive) attributes. It is similar to the *Oblivious Read-Once Decision Graph* (*OODG*) model (Kohavi and Li 1995). "Read-once" means that each nominal feature is tested at most once along any path, which is a common property of most decision-tree algorithms such as C4.5 (Quinlan 1993). The name "oblivious" indicates the fact that all nodes at a given level are labeled by the same feature. The same ordering restriction is imposed by Bryant (1986) on *Function Graphs*. An Information Network has nearly the same structure as an oblivious read-once decision graph with two important differences: it extends the "read-once" restriction of (Kohavi and Li 1995) to continuous features by allowing *multi-way splits* of a continuous domain at the same level and it associates *probability estimates* rather than categorical predictions with each leaf node.

The underlying principle of the Information Network (IN) induction algorithm (Last and Maimon 2004) is to construct a multi-layered network that maximizes the statistically significant Mutual Information (MI) between input and target attributes. Each *hidden* layer is related to a specific input attribute and represents the interaction between this input attribute and those associated with previous layers. The first layer (layer 0) includes only the root node and is not associated with any input attribute. In each iteration, the input attribute having the maximum conditional mutual information is selected by the algorithm resulting in adding a new hidden layer to the network. The Information Network construction algorithm is using a pre-pruning strategy: a node is split if this procedure brings about a statistically significant decrease in the entropy (equal to increase in the mutual information) of the target attribute. This entropy decrease is called "conditional mutual information" between an input attribute and a target attribute. If none of the remaining input attributes provides statistically significant conditional mutual information, the network construction stops.

For each candidate input (predictive) attribute A_i in a layer n, the algorithm calculates the conditional mutual information of A_i and the target (classification) attribute T given n-1 input attributes $X_1, \ldots, X_{n-1}$ by the following formula:

$$MI(T; A_i/X_1, \ldots, X_{n-1}) = \sum_{z \in L_{n-1}} MI(T; A_i/z) \tag{1}$$

where $MI(T; A_i/z)$ is the conditional mutual information of a candidate input attribute A_i with the target attribute T given a terminal node z in the layer $n-1$ (denoted by L_{n-1}). Like in any decision tree, each terminal (leaf) node z of the k-th layer of an Information Network represents a specific conjunction of values of k predictive attributes associated with k hidden layers, respectively. However, due to the read-once nature of Information Networks, the *order* of testing predictive attributes is identical at all leaf nodes.

For nominal predictive attributes, the conditional mutual information of a candidate input (predictive) attribute A_i and the target (classification) attribute T given a node z is calculated by the following formula:

$$MI(T; A_i/z) = \sum_{t=o}^{M_T-1} \sum_{j=0}^{M_i-1} P(C_t; V_{ij}; z) * \log \frac{P(V_{ij}^t/z)}{P(V_{ij}/z) * P(C_t/z)} \tag{2}$$

where

M_T/M_i: number of distinct values ("classes") of the target attribute T / candidate input attribute i, respectively;
$P(V_{ij}/z)$: an estimated conditional probability of a value j of the candidate input attribute i given the node z; it is calculated as the proportion of instances at the node z, where the value of the candidate input attribute i is j.
$P(V_{ij}^t/z)$: an estimated conditional probability of a value j of the candidate input attribute i and a value ("class") t of the target attribute T given the node z; it is calculated as the proportion of instances at the node z, where the value of the candidate input attribute i is j and the value ("class") of the target attribute T is t.
$P(C_t/z)$: an estimated conditional probability of a value ("class") t of the target attribute T given the node z; it is calculated as the proportion of instances at the node z, where the value of the target attribute T is t.
$P(C_t; V_{ij}; z)$: an estimated joint probability of a value ("class") t of the target attribute T, a value j of the candidate input attribute i, and the node z; it is calculated as the proportion of all training instances, where the value of

the candidate input attribute i is j, the value ("class") of the target attribute T is t, and the node is z.

It is important to note that the probability calculations above skip the instances where the values of A_i and/or T attributes are missing.

The conditional entropy of the target (classification) attribute can only be calculated with respect to input attributes taking a finite number of values. For continuous predictive attributes, the algorithm performs discretization "on-the-fly" by recursively finding a binary partition of an input attribute that minimizes the conditional entropy. The conditional mutual information of partitioning an interval S of a candidate input attribute at the threshold Th and the target attribute T given a node z is calculated by the following formula (Last and Maimon 2004):

$$MI(Th; T/S, z) = \sum_{t=o}^{M_T-1} \sum_{y=1}^{2} P(S_y; C_t; z) * \log \frac{P(S_y; C_t/S, z)}{P(S_y/S, z) * P(C_t/S, z)} \quad (3)$$

where,

$P(S_y/S, z)$: an estimated conditional probability of a subinterval S_y, given the partitioned interval S and the node z. It is calculated as the proportion of instances belonging to the partitioned interval S at the node z, where the value of the candidate input attribute belongs to the subinterval S_y. The number of subintervals in each partitioned interval is two.
$P(C_t/S, z)$: an estimated conditional probability of a value ("class") t of the target attribute T given the interval S and the node z; it is calculated as the proportion of instances belonging to the partitioned interval S at the node z, where the value of the target attribute T is t.
$P(S_y; C_t/S, z)$: an estimated joint probability of a value of the target attribute T and a subinterval S_y given the interval S and the node z; it is calculated as the proportion of instances belonging to the partitioned interval S at the node z, where the value of the target attribute T is t and the value of the candidate input attribute belongs to the subinterval S_y.
$P(S_y; C_t; z)$: an estimated joint probability of a value C_t of the target attribute T, a subinterval S_y, and the node z. It is calculated as the proportion of all training instances, where the value of the target attribute T is t, the value of the candidate input attribute belongs to the subinterval S_y, and the node is z.

The main steps of the recursive discretization procedure, initially introduced in (Last and Maimon 2004), are shown in the Appendix.

The statistical significance of the estimated conditional mutual information between a candidate input attribute A_i and the target attribute T given a node z is evaluated by using the following likelihood-ratio statistic:

$$G^2(T; A_i/z) = 2 * (\ln 2) * E^* * MI(T; A_i/z) \quad (4)$$

where E^* is the total number of training cases in the dataset. The null hypothesis is that the actual conditional mutual information is zero and that hypothesis is rejected if the G^2 statistic is significant at the pre-specified confidence level. Based on the empirical results with real-world datasets (Maimon and Last 2000), the default confidence level, leading to the most compact and accurate models, is set to 99.9%, but it can be reduced if larger models involving more predictive features are needed.

The Multi-target Information Network (M-IN)

Product quality is usually measured by multiple dimensions, such as presence/absence of various defects. In order to provide a unified framework for single-target and multi-target classification tasks, Last (2004) has defined an *extended classification task* using the following notation:

- $R = (A_1, \ldots, A_k)$—a set of k attributes in the dataset ($k \geq 2$).
- C—a non-empty subset of n candidate input features ($C \subset R, |C| = n \geq 1$). The values of these features are usually known and they can be used to predict the values of target attributes (see next). Some of candidate input features may represent controllable parameters, which can affect the quality dimensions of the manufactured product.
- O—a non-empty subset of m target ("output") attributes ($O \subset R, |O| = m \geq 1$). This is a subset of attributes representing the variables to predict, such as product quality dimensions. The extended classification task is to build an accurate model (or models) for predicting the values of all target attributes, based on the corresponding dependency subset (or subsets) $I \subseteq C$ of input features.

As shown in (Last 2004), an m-target classification function can be represented by a *multi-target information network* (M-IN), where the nodes of the target layer represent the values of all output attributes. Like a single-target IN, a multi-target information network has a single *root node* and its internal "read-once" structure is identical for all target variables. This means that every hidden node is shared among *all* outputs and each terminal (leaf) node is connected to at least one target node associated with every output.

At every iteration, the M-IN construction algorithm chooses an input (predictive) feature, which maximizes the total conditional mutual information between an input feature

and the subset of output attributes. For each candidate input (predictive) attribute A_i in a layer n, the algorithm calculates the conditional mutual information of A_i and the target (classification) attributes $T_1, \ldots, T_m$ given $n-1$ input attributes $X_1, \ldots, X_{n-1}$ by the following formula:

$$MI(T_1, \ldots, T_m; A_i / X_1, \ldots, X_{n-1}) = \sum_{j=1}^{m} MI(T_j; A_i / X_1, \ldots, X_{n-1}) \quad (5)$$

The remaining calculations are identical to the IN construction algorithm described in the previous sub-section.

According to (Last 2004), the M-IN model has the following information-theoretic properties:

- The average conditional entropy of m target attributes in an n-input m-target model M is not greater than the average conditional entropy over m single-objective models $S_i (i = 1, \ldots, m)$ based on the same n input features. This inequality is strengthened if the multi-objective model M is trained on more features than the single-objective models. Consequently, we may expect that the *average accuracy* of a multi-objective model in predicting the values of m target attributes will not be worse, or even will be better, than the average accuracy of m single-objective models using the same set of input features.
- If all target attributes are either mutually independent or completely dependent on each other, the input feature selected by the M-IN construction algorithm will minimize the joint conditional entropy of all target attributes. This implies that in both these extreme cases, the M-IN algorithm is expected to produce the optimal (most accurate) model.

Optimization with IN and M-IN

In a classification setting, the oblivious decision trees produced by the IN or the M-IN algorithms can be used to predict the values of the target attributes using the majority voting at the terminal node associated with each unlabeled record. The predictive performance of the IN and the M-IN algorithms has been extensively evaluated in Last and Maimon (2004) and Last (2004), respectively. According to the empirical results, the Information Network induction algorithms tend to produce much more compact models than C4.5, while preserving nearly the same level of classification accuracy. This result supports the theorem proven in Bryant (1986) that each Boolean function has a unique function graph representation having a *minimal* number of vertices.

However, even when the predictive capability of the Information Network is limited due to the high noisiness of real-world data, the model structure can still provide us with some useful information about the explored phenomenon. Thus, the user may be interested in the *estimated distribution* of the target attribute across different leaf nodes of the Information Network. For example, in case of a batch manufacturing process, a small increase or decrease in the probability of the "success" class as a result of certain changes in the process settings may lead over time to significant savings in manufacturing costs. The single-target network structure induced for a quality dimension T_j can be used to find the *optimal* process settings z^* corresponding to a leaf node z with the highest probability of a successful outcome:

$$z^* = \arg\max_{z} \Pr(T_j = \text{'success'}/z) \quad (6)$$

In case of m independent quality dimensions represented by a multi-target Information Network, we may be interested to find the settings that minimize the probability of *any* failure using the following equation:

$$z^* = \arg\max_{z} \prod_{j=1}^{m} \Pr(T_j = \text{'success'}/z) \quad (7)$$

Definitely, the above equations aimed at maximizing the success probability can be enhanced with cost considerations as well.

The induction of Information Network models for the crystal manufacturing data will be covered by the next section ("mining crystal quality data"), whereas the search for the optimal process parameters using Eqs. 6 and 7 above will be presented in the section "optimizing the crystal quality with IN models."

Mining crystal quality data

Data description

For this study, we have obtained a database that included 1,289 records of the growth processes performed between July 1995 and November 2005. Each process is described by close to 50 variables. About one-third of those variables are represented by text fields containing code words, special characters (such as '**'), and free text, whereas the remaining variables are stored in a numeric format. Using information provided by the process engineers, all textual fields and some numeric fields have been converted into pre-specified numeric codes. Inconsistent and abnormal attribute values were manually corrected by the process engineers using information from other fields. All missing values were assigned a special missing value code (999999) so that the Information Network algorithm can handle missing values appropriately. After marking several variables as irrelevant to the process outcome, we were left with 45 candidate input (predictive) attributes. These attributes included six controllable variables (process settings) and 39 uncontrollable (random) variables. Following additional data cleaning, we have removed four

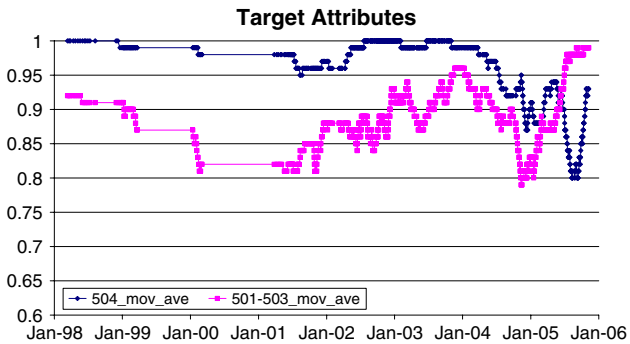

Fig. 1 Temporal behavior of Outcome_501–503 (501–503_mov_ave) and Outcome_504 (504_mov_ave)

records leaving us with 1,285 records of manufactured crystals only.

Our study was aimed at explaining the two most common reasons of process failure denoted by the failure codes "501–503" and "504". Accordingly, we have defined the target variables "Outcome_501–503" and "Outcome_504" as follows:

1) Outcome_501–503 = 0 (failure) if the failure code is 501, 502 or 503; otherwise Outcome_501–503 = 1 (success). This defect was found in 10.1% of crystals (130 out of 1,285).
2) Outcome_504 = 0 (failure) if the failure code is 504; otherwise Outcome_504 = 1 (success). This defect was found in 4.7% of crystals (60 out of 1,285).

The temporal behavior of both target attributes is shown in Fig. 1, which presents the moving averages of Outcome_501–503 (*501–503_mov_ave*) and Outcome_504 (*504_mov_ave*) over a sliding window of 100 observations. In other words, each point on the chart represents the portion of successful outcomes out of the 100 most recent batches. The process engineers were particularly interested to explain the sharp decrease in the process quality, in terms of both target attributes, which took place in the course of 2004. Equally important for them was to understand the reasons for a gradual improvement of Outcome_501–503, which continued since the end of 2004, along with the apparent instability of Outcome_504 during the same period. The most recent improvement in Outcome_504, which started around August 2005, also required an explanation.

Induction of predictive models

First, we have attempted to find models explaining the variability in both target attributes by running the Information Network algorithm on *all* 45 predictive attributes. However, the selected input features and the induced rules were judged as useless by the process engineers, since they mainly indicated the well-known relationship between the production date and the process failure rate without providing any clues to improving the settings of the manufacturing process. Consequently, we have decided to focus on the six controllable variables only. We have also learned that these six parameters can in no way be treated as "random variables", since they are changed by process engineers quite infrequently (usually not more than once in a year). Thus, we needed to find out, which parameters were less stable (more time-dependent) during the period of study.

The effect of time on each controllable variable (denoted in this paper by the letters S, E, O, M, N, and C) was evaluated using the single-target Information Network algorithm by defining each controllable variable as a target attribute and the FDATE (completion date) variable as the only input attribute. The default confidence level of the IN algorithm (99.9%) remained unchanged. The ranges of continuous target attributes (S, E, O, and M) were discretized into three intervals of equal frequency. The results of the corresponding six runs of the IN algorithm are presented in Table 1. Each table row shows the name of a controllable variable, the number of its discretization intervals or nominal values, its unconditional entropy, the mutual information between the FDATE variable and the controllable variable, and the ratio of the mutual information to the total entropy of the controllable variable. A higher ratio indicates a stronger effect of time on the controllable variable, since it means that the time can explain a higher portion of the variable entropy (uncertainty). Though the effect of time on all six variables

Table 1 The effect of time on controllable Variables

Target attribute	Number of intervals/values	Total entropy	Mutual Information (*MI*)	*MI/entropy %*
S	3	0.979	0.905	92.44
C	2	0.972	0.850	87.45
E	3	1.554	0.482	31.02
O	3	1.557	0.139	8.73
M	3	1.200	0.087	7.25
N	4	1.792	0.085	4.74

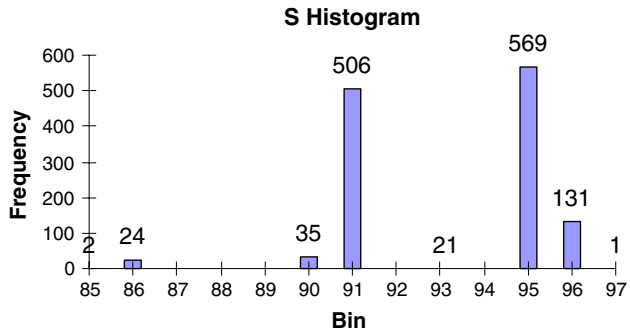

Fig. 2 Distribution of the S variable

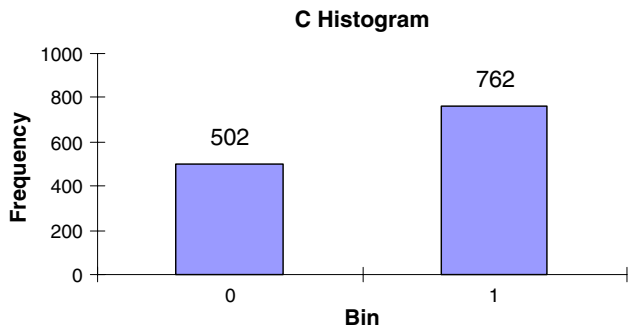

Fig. 3 Distribution of the C variable

was found statistically significant by the algorithm, the first two variables (S and C) were apparently subject to much more frequent changes than the other four variables, causing a considerable decrease in their entropy (nearly 90%) given the FDATE attribute.

The probability distributions of S and C are shown in Figs. 2 and 3, respectively. C is a discrete binary variable, which can be set to either 0 or 1, whereas S is a continuous variable taking integer values between 85 and 96. The two most common values of S were 91 and 95 though sometimes it was set to seven other values in its range such as 96. C was assigned the level of zero in nearly 40% of all processes (502 records) and the level of one in the remaining 60% (762 records). The reliability of all recorded values was verified by the process experts.

The moving averages of C and S as a function of time are shown in Fig. 4. On the S curve, one can identify a continuous increase until May 2002, stability from May 2002 to July 2003, and then a decrease back to the initial level except for a brief increase after May 2004. On the other hand, C, which is a discrete binary variable, was set to 1 between July 2002 and February 2003, kept at the level of 1 until July 2005, and then reset back to 0.

To find the exact effect of both controllable variables (S and C) on each failure type, we have run the IN algorithm separately for every target attribute defined above. In the case of Outcome_501–503, the default confidence level of 99.9% has produced a model containing only one input attribute (S). The level reduction to 99.0%, has added the second input attribute (C) to the model. The detailed results of each iteration of the IN algorithm are shown in Table 2. The mutual information (MI) between the first input variable (S) and the target Outcome_501–503 is 0.014 bits leaving the conditional entropy of the target at 0.459 bits. The second input variable (C) is reducing the conditional entropy of the target by additional 0.004 bits (see the "Conditional MI" column). The total mutual information of this model is very small (0.018 bits) compared to the total entropy of the target attribute (0.473 bits). This implies that S and C can only explain a small portion of the Outcome_501–503 uncertainty.

The structure of the induced Information Network is shown in Fig. 5. The two network hidden layers represent the first input attribute (S) and the second input attribute (C), respectively. Connections representing input attribute values, such as "S $<$ 94", are denoted by dotted lines, whereas connections between terminal nodes (1, 3, 4, and 5) and the nodes of the target layer (Failure/Success) are shown as solid lines. Each terminal-target connection is accompanied by the probability of the corresponding outcome (Failure/Success). Thicker lines represent the most likely (majority) outcome at each terminal node. On the other hand, the prediction model induced for the second target attribute (Outcome_504) contains both input attributes for the default confidence level of 99.9%. The detailed results of each iteration of the IN algorithm are shown in Table 3. The mutual information (MI) between the first input variable (S) and the target Outcome_504 is 0.018 bits leaving the conditional entropy of the target at 0.254 bits. The second input variable (C) is reducing the conditional entropy of the target by additional 0.011 bits (see the "Conditional MI" column). The total mutual information (uncertainty reduction) of the obtained model is 0.029 bits, which is still small compared to the unconditional entropy of the target (0.272), but much higher than the MI of the Outcome_501–503 model (0.018 bits).

The structure of the induced Information Network is shown in Fig. 6. The two network hidden layers represent the first input attribute (S) and the second input attribute (C), respectively. Like in Fig. 5, connections representing input attribute values, such as "S $<$ 94", are denoted by dotted lines, whereas connections between terminal nodes (2, 3, and 4) and the nodes of the target layer (Failure/Success) are shown as solid lines. Each terminal-target connection is accompanied by the probability of the corresponding outcome (Failure/Success) with thicker lines representing the most likely (majority) outcome at each terminal node.

The optimal process conditions based on the induced models are explored in the next section.

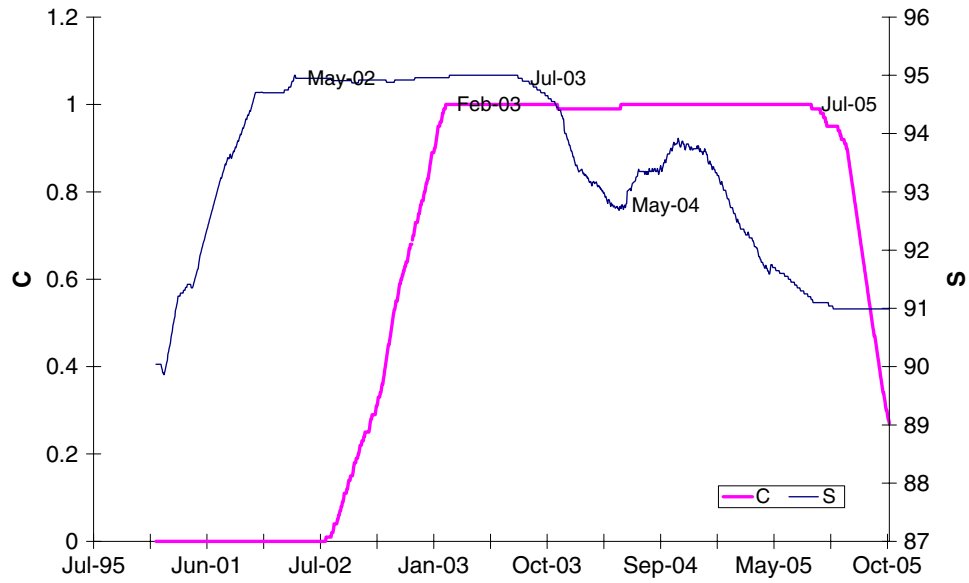

Fig. 4 Moving averages of S and C as a function of time

Table 2 IN run summary for target = Outcome_501–503

Iteration	Attribute name	Mutual Information (MI)	Conditional MI	Conditional entropy	Split nodes
0	S	0.014	0.014	0.459	1
1	C	0.018	0.004	0.455	1

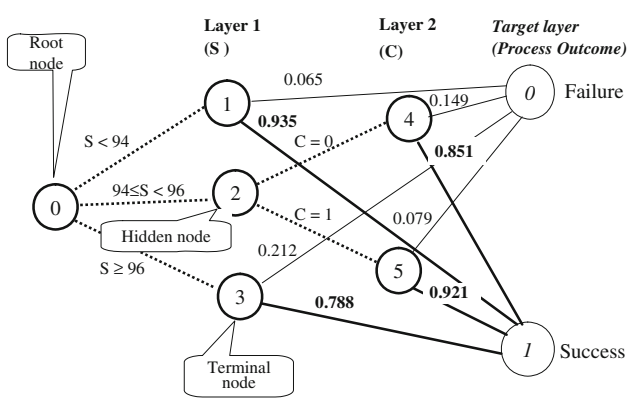

Fig. 5 Information Network for target = Outcome_501–503

Table 3 IN run summary for target = Outcome_504

Iteration	Attribute name	Mutual Information (MI)	Conditional MI	Conditional entropy	Split nodes
0	S	0.018	0.018	0.254	1
1	C	0.029	0.011	0.243	1

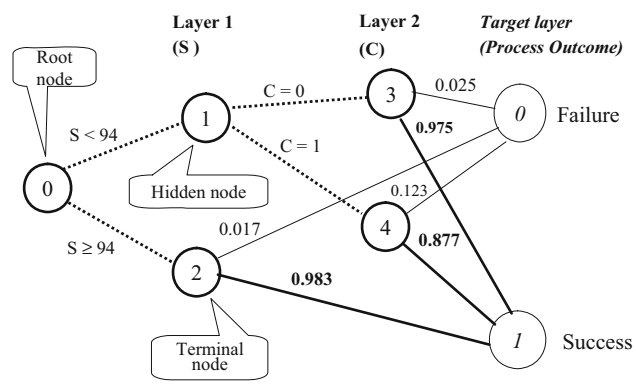

Fig. 6 Information Network for target = Outcome_504

Optimizing the crystal quality with IN models

Table 4 shows the estimated distribution of the target attribute Outcome_501–503 across different leaf nodes of the Information Network presented in Fig. 5 above. By applying Eq. 6 to the data in Table 4, we can conclude that the lowest failure rate of 6.5% for failures 501–503 is achieved if the *S parameter is kept below 94, disregarding the value of C*. On the other hand, raising S to the value of 96 and higher should triple the failure rate to the level of 21.2%.

Based on the probability estimation rules in Table 4 and moving averages of S and C in Fig. 4, we can finally explain the variability in the 501–503 failure rates between the years 1995-2005 as observed in Fig. 1 above:

- The average value of S was gradually increased from 90 to 95 between July 1998 and May 2002, whereas C was kept at the level of zero. Based on *Rules 0 and 2*, this caused an increase in the failure rate from 6.5% to 14.9%.

Table 4 Estimated distribution of the target attribute Outcome_501–503

Probability estimation rule No.	Rule condition	Prob. (501–503 = 0)	Prob. (501–503 = 1)
0	If S is between 0 and 94	0.065	0.935
1	If S is 96 and higher	0.212	0.788
2	If S is between 94 and 96 and C is 0	0.149	0.851
3	If S is between 94 and 96 and C is 1	0.079	0.921

Table 5 Estimated distribution of the target attribute Outcome_504

Probability estimation rule No.	Rule condition	Prob. (Outcome = 0)%	Prob. (Outcome = 1)%
1	S is 94 and higher	1.7	98.3
2	S is between 0 and 94 and C is 0	2.5	97.5
3	S is between 0 and 94 and C is 1	12.3	87.7

- The average value of S was kept close to 95 until July 2003, but between July 2002 and February 2003 C was raised to the level of one. Based on *Rule 3*, this change reduced the failure rate to 7.9%.
- Between January 2004 and March 2005 some crystals were grown with S = 96 resulting in an increase of S average value. Based on *Rule 1*, this change caused a drastic drop in the crystals quality represented by a failure rate of 21.2%.
- Starting from May 2005 nearly all crystals were grown with S = 91. Based on *Rule 0*, this change restored the former minimal failure rate of 6.5%.

Table 5 shows the estimated distribution of the target attribute Outcome_504 across different leaf nodes of the Information Network presented in Fig. 6 above. By applying Eq. 6 to the data in Table 5, we can conclude that the lowest failure rate of 1.7% for failure 504 is achieved if the *S parameter is kept at the level of 94 and higher*, *disregarding the value of C*. On the other hand, decreasing S below the value of 94 and keeping C at the level of one should increase the failure rate to the level of 12.3%, which is more than seven times higher (!) than the lowest achievable failure rate.

Based on the probability estimation rules in Table 5 and moving averages of S and C in Fig. 4, we can also explain the variability in the 504 failure rates between the years 1995–2005 as observed in Fig. 1 above:

- Before May 2002 the average value of S was low (below 94), whereas C was kept at the level of zero. Based on *Rule 2*, these conditions resulted in a relatively low failure rate of 2.5%.
- By mid-2002 the average value of S was raised above 94. Based on *Rule 1*, this did not harm the average quality and even slightly improved it by reducing the failure rate from 2.5% to 1.7%.
- At the end of 2002—beginning of 2003, the C level was raised to one without changing the value of S. Based on *Rule 1*, this change did not affect the average crystal quality.
- In the second half of 2003—beginning of 2004, S was restored to its former level (below 94) *without changing* the C level of one. *Based on Rule 3*, this change caused a dramatic increase in the failure rate from 1.7% to 12.3%.
- In July 2005, C was reset to the level of zero, which was maintained before mid-2002. Based on *Rule 2*, this change returned the failure rate to the low level of 2.5%, which was experienced before 2002.

For natural reasons, the process engineers are interested in reducing *all* kinds of defects including failures 501–503 *and* 504. However, there is a clear conflict between the recommendations of the two models induced above. The condition that minimizes the rate of 504 failures ("S of 94 and higher") contradicts the recommendation of the Outcome_501–503 model ("S below 94"). To find the optimal process settings that take into account both failure types, we have run the multi-target Information Network (M-IN) algorithm to estimate simultaneously the distributions of two target attributes: Outcome_501–503 and Outcome_504. The default confidence level (99.9%) remained unchanged. The run summary and the induced probability estimation rules are shown in Tables 6 and 7, respectively.

The last column of Table 7 calculates the probability of *any* of the two failure types based on Eq. 7, which assumes that these are two independent events. The independency assumption was confirmed by the process engineers with respect to failures 501–503 vs. 504. Based on Table 7, *Rule 1* provides the optimal process settings, which minimize the probability of both failure types:

S is between 0 and 94 and C is 0

Table 6 M-IN run summary for targets = Outcome_501–503 and Outcome_504

Iteration	Attribute Name	Total Mutual Information (MI)	Total conditional MI	Total conditional entropy	Split nodes
0	S	0.027	0.027	0.718	1
1	C	0.038	0.011	0.707	1

Table 7 Estimated distributions of the target attributes Outcome_501–503 and Outcome_504

Probability estimation rule No.	Rule condition	Prob. (504 = 0)	Prob. (504 = 1)	Prob. (501–503 = 0)	Prob. (501–503 = 1)	Prob. (501–504)
0	If S is 94 and higher	0.017	0.983	0.131	0.869	0.146
1	If S is between 0 and 94 and C is 0	0.025	0.975	0.08	0.92	0.103
2	If S is between 0 and 94 and C is 1	0.123	0.877	0.056	0.944	0.172

Under the above conditions, the expected failure rate is 10.3% only.

We have also explored the potential benefit of this data mining analysis if it were applied to the collected data at *an earlier date*. It turned out that very similar recommendations could be extracted from a multi-target IN model using the 99.0% confidence level and 759 records collected until December 2003. The discovery of the optimal process settings in December 2003 would probably keep the overall failure rate between January 2004 and December 2005 at the low level of 10.3% instead of the actual failure rate of 18.8% during the same period.

Conclusions

In this study, we have demonstrated that probability estimation models induced by Information Network algorithms can be successfully utilized to determine the optimal process settings of a complex manufacturing process such as crystal growth. For an effective implementation of the proposed methodology, it is very important to identify the controllable parameters that are frequently changed by process engineers. Several independent quality dimensions can be taken into account using multi-target Information Network models. We have also shown that a smaller dataset collected at an earlier date could improve the past outgoing quality in terms of the failure rate.

We believe that process optimization with probability estimation models can be further explored in several directions. These include application of alternative data mining algorithms, estimation of minimal required sample sizes, online change detection, cost-sensitive optimization, and many others. It is also important to keep in mind that Information Networks, due to their "oblivious" nature, are restricted by a constant order of attribute tests along each path. Consequently, we may lose some potentially more accurate rules, where the set and the order of tested attributes are different. The effect of this limitation on the Information Network predictive accuracy was studied extensively in (Last and Maimon, 2004).

Acknowledgements This work was partially supported by a research grant from Israel Ministry of Defense.

Appendix: discretization of continuous attributes with IN algorithm

Partition (Data Table r, Information Network, Attribute A_i, Interval S, Significance level sign)

Input: the set of n training instances, an information-theoretic network, a continuous attribute A_i to be discretized, the interval S to be partitioned (the first and the last distinct values of A_i), and the minimum significance level *sign* for splitting an interval (default: $sign = 0.1\%$).

Output: the list of threshold values for A_i

Step 1— For every distinct value *Th* included in the interval S (except for the last distinct value) Do:

- *Step 1.1*— For every node z of the final hidden layer Do:
  - *Step 1.1.1*— Calculate the likelihood-ratio test for the partition of the interval S at the threshold *Th* and the target attribute T given the node z. All values below or equal to *Th* belong to the first sub-interval S_1. Distinct values above *Th* belong to the second sub-interval S_2.
  - *Step 1.1.2*— If the likelihood-ratio statistic is significant, mark the node as "split" by the threshold *Th*
  - *Step 1.1.3*— End Do
- *Step 1.2* — End Do

Step 3— Find the threshold $Th_{\max}$ maximizing the conditional mutual information over all nodes

Step 4— If the maximum estimated conditional mutual information is greater than zero, then Do:

Step 4.1— For every node z of the final hidden layer Do:

Step 4.1.1— If the node z is split by the threshold $Th_{\max}$, mark the node as split by the candidate input attribute A_i

Step 4.2— If the threshold Th_{max} is the first distinct value in the interval S, mark Th_{max} as the lower bound of a new discretization interval, else *Partition (Data Table r, Network, Attribute* A_i, *Interval* S_1, *sign*).

Step 4.3— *Partition (Data Table r, Network, Attribute* A_i, *Interval* S_2, *sign*)

Step 4.4— End Do

Step 5— Else return the list of threshold values for A_i

More details are provided in (Last and Maimon 2004).

References

Babu, J., & Frieda, W. H. (1993). Predictive control of quality in a batch manufacturing process using artificial neural network models. *Industrial & Engineering Chemistry Research, 32*(9), 1951–1961. doi:10.1021/ie00021a019.

Biderman, S., Horowitz, A., Einav, Y., Ben Amar, G., Gazit, D., Stern, A., et al. (1991). Production of sapphire domes by the growth of near-net-shape single crystals. In *Proceedings of SPIE 1535* (*Passive Materials for Optical Elements*) (pp. 27–34).

Braha, D., Elovici, Y., & Last, M. (2007). Theory of actionable data mining with application to semiconductor manufacturing control. *International Journal of Production Research, 45*(13), 3059–3084. doi:10.1080/00207540600654475.

Bryant, R. E. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers, C-35-8*, 677–691.

Cook, D. F., Ragsdale, C. T., & Major, R. L. (2000). Combining a neural network with a genetic algorithm for process parameter optimization. *Engineering Applications of Artificial Intelligence, 13*, 391–396. doi:10.1016/S0952-1976(00)00021-X.

Famili, A. (1994). Use of decision-tree induction for process optimization and knowledge refinement of an industrial process. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing. (AI EDAM), 8*(1), 63–75.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques* (2nd ed.). San Francisco, CA: Morgan Kaufmann.

Horowitz, A., Biderman, S., Ben Amar, G., Laor, U., Weiss, M., & Stern, A. (1987). The growth of single crystals of optical materials via the gradient solidification method. *Journal of Crystal Growth, 85*(1–2), 215–222. doi:10.1016/0022-0248(87)90225-9.

Horowitz, A., Biderman, S., Gazit, D., Einav, Y., Ben Amar, G., & Weiss, M. (1993). The growth of dome shaped sapphire crystals by the GSM method. *Journal of Crystal Growth, 128*(1–4 pt 2), 824–828.

Hur, J., Lee, H., & Baek, J.-G. (2006). An intelligent manufacturing process diagnosis system using hybrid data mining. In *Advances in Data Mining* (pp. 561–575). Springer-Verlag.

Kohavi, R., & Li, C.-H. (1995). Oblivious decision trees, raphs, and top-down pruning. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1071–1077).

Last, M. (2004). Multi-objective classification with info-fuzzy networks. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*. Lecture Notes in Artificial Intelligence 3201 (pp. 239–249). Springer-Verlag.

Last, M., & Kandel, A. (2001). Data mining for process and quality control in the semiconductor industry. In D. Braha (Ed.), *Data mining for design and manufacturing: Methods and applications*. Kluwer Massive Computing Series (Vol. 524, pp. 207–234). Norwell, MA: Kluwer Academic Publishers.

Last, M., & Maimon, O. (2004). A compact and accurate model for classification. *IEEE Transactions on Knowledge and Data Engineering, 16*(2), 203–215. doi:10.1109/TKDE.2004.1269598.

Liau, L. C.-K., Yang, T. C.-K., & Tsai, M.-T. (2004). Expert system of a crude oil distillation unit for process optimization using neural networks. *Expert Systems with Applications, 26*(2), 247–255. doi:10.1016/S0957-4174(03)00139-8.

Lin, J. L., & Lin, C. L. (2005). The use of grey-fuzzy logic for the optimization of the manufacturing process. *Journal of Materials Processing Technology, 160*(1), 9–14. doi:10.1016/j.jmatprotec.2003.11.040.

Maimon, O., & Last, M. (2000). *Knowledge discovery and data mining —the Info-Fuzzy Network (IFN) methodology*. Boston: Kluwer Academic Publishers, Massive Computing.

Myers, R. H., & Montgomery, D. (2002). *Response surface methodology: Process and product optimization using designed experiments* (2nd ed.). John Wiley and Sons.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Quinlan, J. R. (1993) *C4.5: Programs for machine learning*. Morgan Kaufmann.