Data Preparation Techniques for Improving Rare Class Prediction

KITTISAK KERDPRASOP and NITTAYA KERDPRASOP
Data Engineering Research Unit,
School of Computer Engineering, Suranaree University of Technology,
111 University Avenue, Nakhon Ratchasima 30000
THAILAND
kittisakThailand@gmail.com, nittaya@sut.ac.th

Abstract: - Rare class prediction is the data mining task aiming at building a model that can correctly identify objects or events rarely occurring in the data set. In many real life applications such as identification of intruders accessing a network system, detecting fraudulent credit card transactions, it is rare events that are of great interest. Unfortunately, traditional mining algorithms fail to predict rare events because the model are inherently built in favor of the majority class to draw common characteristics among data instances. Rare class mining is thus a challenging problem in some specific domains. We study the rare class mining problem in the context of semiconductor manufacturing process control in which fault products are rarely occurred, but once occurring they require timely detection to prevent the decrease in product yield. In this paper, we propose to use an over-sampling technique to alleviate the outnumber situation of majority class. Such sampling technique is however prone to introducing the over-fitting problem. We thus propose the remedy by applying the cluster based technique to selectively extract data instances showing discrimination characteristics. The built models from various mining algorithms have been tested with a separate data set and the results show significant improvement on the predicting accuracy.

Key-Words: - Over-sampling, Data mining, Rare class mining, Feature selection, Manufacturing process.

1 Introduction

Data mining is about the application of learning algorithm to build model that can best characterize underlying data and accurately predict the class of unlabelled data. The quality of data mining model depends directly on the quality of the training data. Data of low quality are those that contain noise, missing values, and class imbalance. A data set is imbalanced if the number of data instances in one class is much more than those in other classes. In the presence of class imbalance, data mining models are biased toward the majority class in such a way that the models can predict the majority class correctly but data instances from the minority class tend to be incorrectly predicted. This is a problem known as rare class mining. This research issue has recently received much attention from the data mining and machine learning research community [1], [4], [7], [15], [18], [19].

In the context of data mining, rare class refers to labeled data instances that are infrequently occurred in the database. Specific problems, such as customer churn prediction [3], network intrusion detection [7], discovering fault software modules [11], are more interested in modeling infrequent patterns than the frequent ones.

To solve the problem of biased learning toward the majority class, many researchers consider the sampling techniques for manipulating class distribution such that rare class could be sufficiently represented in the training data. The basic sampling techniques that have been applied are under-sampling and over-sampling. Under-sampling [16] alters the class distribution by removing data instances from the minority class, whereas over-sampling [1], [2], [13] duplicates data instances in the minority class.

Although sampling methods are simple and yet efficient for mining rare objects, under-sampling may remove good representatives, while over-sampling may cause the over-fitting problem. In this paper, we propose the unsupervised feature selection technique to be applied to the training data prior to the application of over-sampling technique duplicating the rare class instances to the same proportion to the majority class. Our experimental studies on manufacturing process data set yield satisfactory results in that the proposed method can induce the accurate models for predicting both majority and minority test data instances without incurring the over-fitting problem.
2 Intelligent Methods for Predicting Failure in Manufacturing Process

Process control is crucially important to the semiconductor industries that operate the multistage manufacturing systems on the product scale of lesser 300 nanometers [10]. Modern technology in semiconductor manufacturing enables real time process control with the measurement data obtained from the equipment sensors and the final electrical test. With such high volume of data recorded during the entire production process, effective monitoring and optimal process control by investigating and analyzing these data are difficult work for process engineers. Traditional process control methodology like univariate and multivariate control charts is no longer an efficient method to control manufacturing systems with hundreds of processing stages. Instead automatic and advanced process control method is required.

Ison and colleagues [9] proposed a decision tree classification model to detect fault of plasma etch equipment. The model was built from the five sensor signal data. Many researchers also studied the fault detection problem during the etch process. Goodlin et al [6] proposed to build a specific control chart for detecting specific type of faults. They collected tool-state data directly from the etcher. These data consist of 19 variables. The work of Spitzlsperger and colleagues [12] was also based on the statistical method. They adopted the multivariate control chart method to maintain changes in the mean and standard deviation coefficients by remodelling technique.

Recent interest in fault detection has been shifted toward the non-parametric approaches. He and Wang [8] proposed to use the k-nearest neighbor rule for fault detection. Verdier and Ferreira [17] also applied the k-nearest neighbor method, but they proposed to use the adaptive Mahalanobis distance instead of the traditional Euclidean distance. Tafazzoli and Saif [14] proposed a combined support vector machine methodology for process fault diagnosis. Ge and Song [5] applied support vector data to the principal component analysis method to detect process abnormalities.

Most work on fault detection methods has studied the process control problem with a few features of tool-state and process-state measurement data. In our work, we analyze the wafer fabrication data collected from 590 sensors with the last feature is a label stating pass or fail state. The observed data contain 1,463 pass cases with only 104 fail cases. We devise data preparation techniques to deal with highly imbalance between the pass and fail cases.

3 Over-sampling Method on Selective Features

We analyze the wafer fabrication data, called SECOM data set (Semi COnductor Manufacturing, available at www.causality.inf.ethz.ch/repository.php). The dataset is in a form of matrix; rows represent each instance and columns represent features which are values recorded from each sensor. The dataset contains 1567 data instances or examples taken from a wafer fabrication production line. Each example is a vector of 590 sensor measurements plus a label of pass/fail test. Among the 1567 examples, there are only 104 fail cases which are labelled as positive (encoded as 1), whereas much larger amount of 1463 examples pass the test and are labelled as negative (encoded as -1).

The imbalance of pass and fail examples in addition to the large number of metrology data obtained from hundreds of sensors make this dataset a difficult one to accurately analyze. It is thus our main focus to devise a data preparation method to build an accurate model for fault detection.

Firstly, we devise a feature selection technique to extract only prominent variables appropriate for the mining step. Our feature selection technique is based on a clustering technique. Next, we apply the over-sampling technique to duplicate fail cases to the same amount of pass cases. On the model building phase, we apply four methods to induce the fault-detection model namely decision tree, naïve Bayes, logistic regression, and k-nearest neighbour algorithms. The steps in our data preparation and mining phase are shown as algorithm in Figure 1.

We assess the model performance based on the three metrics: true positive rate (recall), precision, and F-measure. The computation methods of these metrics are as follows.

\[
\text{TP rate (or Recall, Sensitivity)} = \frac{TP}{TP + FN}
\]

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

\[
\text{F - measure} = \frac{2TP}{2TP + FP + FN}
\]

where

TP = number of fail cases that are correctly identified as fail,
FP = number of pass cases that are incorrectly identified as fail cases,
FN = number of fail cases that are misclassified as pass cases, and
TN = number of pass cases that are correctly identified as pass cases.
Data Cleaning Phase
(1) Investigate data observed from each sensor, i.e. data in each column. If the data appear to be a single value, then remove that feature.
(2) Count in each column the ‘not available’ or missing values. If data are missing more than 55%, then remove that feature.

Feature Selection Phase
(3) Apply the following cluster-based feature selection technique:
(3.1) Clustering data into two clusters (fail cluster and pass cluster)
(3.2) Compare value differences in every feature of the fail cluster mean and the pass cluster mean
(3.3) Ranking features in descending order according to the magnitude of mean differences computed in step 3.2, and output the ranked features

Over-sampling Phase
(4) Separate data obtained from step 2 into two datasets: train data and test data. Each data set maintains the same proportion of pass and fail cases.
(5) Increase the number of fail cases in the train data by duplicating the fail cases to be the same amount as the pass cases.

Model Building Phase
(6) Build a prediction model with decision tree, naïve Bayes, k-nearest neighbor, and logistic regression algorithms.
(7) For datasets from steps 3, evaluate model accuracy with 10-fold cross validation technique. Dataset from step 5 is evaluated with the test set.

4 Experimentation and Results
We use the Weka software (www.cs.waikato.ac.nz/ml/weka) to perform a series of experiments. The first part of our study aims at selecting principal features that show the most discrimination power of differentiating fail cases from pass cases.

In the cleaning step, we remove 137 features that contain a single value or a lot of missing values. From the remaining 454 features, we select the best 168 features (to maintain around 95% of variances) by means of cluster-based method (feature selection phase in Figure 1). The fault detection models are then derived from each feature selected data. We want the model that shows the highest values of TP rate, precision, and F-measure.

The next step is the separation of SECOM dataset into a train set and a test set. The test set contains 468 instances in which 59 instances are fail cases and 409 are pass cases. The train set contains 45 instances of fail cases and 1054 of pass cases.

We then duplicate the number of fail cases in the training data to be 1096 instances. The fault detection models are built from this rare case over-sampling training dataset. The models are then evaluated their classification performances by the separated test dataset. The classification error matrices of models built from the four different learning methods are given in Figure 2. The true positive rate, precision, and F-measure of each model are also graphically provided in Figure 3.

<table>
<thead>
<tr>
<th>k-Nearest Neighbor:</th>
<th>Predicted class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td>Class=1 (fail)</td>
</tr>
<tr>
<td>Class= 1</td>
<td>58</td>
</tr>
<tr>
<td>Class= -1</td>
<td>98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logistic regression:</th>
<th>Predicted class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td>Class=1 (fail)</td>
</tr>
<tr>
<td>Class= 1</td>
<td>59</td>
</tr>
<tr>
<td>Class= -1</td>
<td>137</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Naïve Bayes:</th>
<th>Predicted class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td>Class=1 (fail)</td>
</tr>
<tr>
<td>Class= 1</td>
<td>44</td>
</tr>
<tr>
<td>Class= -1</td>
<td>144</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision Tree:</th>
<th>Predicted class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td>Class=1 (fail)</td>
</tr>
<tr>
<td>Class= 1</td>
<td>59</td>
</tr>
<tr>
<td>Class= -1</td>
<td>66</td>
</tr>
</tbody>
</table>

Figure 1. Data preparation and model building algorithm.

Figure 2. Performance metrics of each fault detection model.
conclude that decision tree model is a good candidate for automatic generation of the fault detection model to be used in the semiconductor manufacturing process. The fault detection model in a form of decision tree is given in Figure 4.

$S_{511} \leq 28.3784$: Predict -1
$S_{511} > 28.3784$
$S_{470} \leq 4.3751$: Predict -1
$S_{470} > 4.3751$
$S_{16} \leq 423.3311$
$S_{16} > 401.1307$
$S_{472} \leq 4.4751$: Predict -1
$S_{472} > 4.4751$
$S_{51} \leq 646.9073$
$S_{51} > 646.9073$
$S_{4} \leq 905.1501$: Predict -1
$S_{4} > 905.1501$
$S_{188} \leq 11.54$: Predict -1
$S_{188} > 11.54$
$S_{431} \leq 3.8926$: Predict -1
$S_{431} > 3.8926$
$S_{439} \leq 28.6219$: Predict -1
$S_{439} > 28.6219$
$S_{495} \leq 1.3638$
$S_{495} > 1.3638$
$S_{56} \leq 2875$
$S_{56} > 2875$
$S_{548} \leq 398.552$: Predict -1
$S_{548} > 398.552$
$S_{178} \leq 0.448$
$S_{178} > 0.448$
$S_{29} \leq 73.4556$
$S_{29} > 73.4556$
$S_{578} \leq 16.4303$
$S_{578} > 16.4303$
$S_{414} \leq 25.0931$: Predict -1
$S_{414} > 25.0931$
$S_{161} \leq 614$: Predict -1
$S_{161} > 614$
$S_{273} \leq 19.8922$: Predict -1
$S_{273} > 19.8922$
$S_{28} \leq 7.373$: Predict -1
$S_{28} > 7.373$
$S_{495} \leq 1.3638$: Predict -1
$S_{495} > 1.3638$
$S_{51} \leq 646.9073$: Predict -1
$S_{51} > 646.9073$
$S_{16} \leq 423.3311$: Predict -1
$S_{16} > 423.3311$

Figure 3. Significant high increases in TP rate (or recall), Precision, and F-measure of fault detection models from applying the cluster-based feature selection and over-sampling techniques.

It can be seen from the graphical comparison as shown in Figure 3 that our data preparation techniques that combine the cluster-based feature selection and the over-sampling strategy to deal with class imbalance can significantly improve the recall, precision, and f-measure of the four learning methods. Among the four algorithms, decision tree learning is the most accurate one. We thus can conclude that decision tree model is a good candidate for automatic generation of the fault detection model to be used in the semiconductor manufacturing process. The fault detection model in a form of decision tree is given in Figure 4.

Figure 4. A decision tree model for fault-detection in the semiconductor process control.
The top level of the decision tree is on the left hand side in which the value from sensor number 511 is the first parameter to be considered. The normal state (encoded as -1) is expected if the value of sensor 511 is less than or equal 28.3784. The fault state is to be detected when the following sensor values are reported: S511 > 28.3784, S470 > 4.3751, S16 > 401.1307, S472 > 4.4751, S51 ≤ 646.9073, S4 > 905.1501, S188 > 11.54, S431 > 3.8926, S439 > 28.6219, S495 > 1.3638 S56 > 2875, S548 > 398.552, S178 ≤ 0.448, S29 ≤ 73.4556, S578 ≤ 16.4303, S474 ≤ 27.9511, and S39 ≤ 86.3506. Other prediction rules can be interpreted in the same manner.

5 Conclusion
The problem of predicting correctly rarely occurring cases is important in many real life applications including the process control of semiconductor manufacturing. In such application, hundreds of metrology data are available for process engineers to analyze for the purpose of maintaining efficient operations and getting optimum yield of high quality products. For such a large volume of measurement data with imbalanced cases of normal versus failure products, automatic and accurate fault detection technique is essential.

We thus investigate the application of data preparation techniques including cluster-based feature selection and over-sampling to create a new training data set of balanced classes. Then apply data mining techniques such as decision tree induction, naïve Bayes analysis, logistic regression, and k-nearest neighbor classification for creating an accurate model for fault case detection in the wafer fabrication process of semiconductor industries.

From a series of experimentation, we found that naïve Bayes model built from a subset of features can detect the fault cases at the very high rate. But the false alarm rate, or false positive, is also high as well. The decision tree method built from our cluster-based feature selection method generates a more comprehensible form of fault detection model with false alarm rate at only 4.5%. But the precision and true positive rate, or recall, of the tree model are still low at 20.5% and 16%, respectively.

We then apply an over-sampling technique to improve the precision of tree-based model for fault detection by duplicating the number of rare cases, or fault test, to the equal number of majority cases, or pass test. The outcome is surprising that the true positive rate of the tree-based model can increase up to 100%, whereas the false alarm rate is still low at the 16%. We plan to investigate our techniques to other domains that show imbalance among data classes in our future research.

Acknowledgment
This work has been supported by grants from the National Research Council of Thailand (NRCT) and Suranaree University of Technology via the funding of Data Engineering Research Unit.

References:


