STANFORD UNIVERISTY

MACHINE LEARNING CS229

Early defect identification of semiconductor processes using machine learning

Friday, December 16, 2011

Authors:
Saul ROSA
Anton VLADIMIROV

Professor:
Dr. Andrew NG.

Friday, December 16, 2011

Early defect identification of semiconductor processes using machine learning

Saul Rosa, Anton Vladimirov

***Abstract*—Early wafer defect identification can account for significant savings in test time and assist in improving the fabrication process. Defect measurements with exhaustive parametric structures can be extremely costly and prohibitive. Embedded rapid parametric data collection, using VCOs, has been added to the chip design, at the cost of semiconductor real estate. So far the data has been considered too noisy and unreliable to draw conclusions off of. Using machine learning, models can be generated, which allow for proper classification of chip quality. This paper presents a method of applied machine learning to take advantage of this data.**

I. Introduction

In chip manufacturing processes and tools, generally have a large impact on the final quality of chips. Due to the time intensive nature of measuring parameter structures that indicate the quality of the tools and processes at each stage only a sample of wafers are measured. As the production line matures, less sample wafers are selected. This creates an issue of being unable to track production process quality over the lifespan of chips. Manufacturers have attempted to solve this issue by embedding dummy measurement structures that have no functional purpose other than to measure these process changes. Due again to test time restrictions these structures have limitations on the methods used to measure device effects.

Most of these structures rely on voltage controlled oscillator (VCO) methods of measuring singular effects. Since each value is only directly related to one device effect the values often indicate little about the chip quality. Values that show significant shift are often readily caught by the fab without further testing. However, combination of parameter values may give us more insight into overall chip quality. There is no known algorithm to determine which combinations of parameters are indicative of overall quality. Instead we use machine-learning strategy to discover them.

In the design selected for this study there are 50 selected parameters and each has dependencies on the other. Some of the parameters may have little effect while others have a strong correlation to the overall quality of a chip. Given a detailed knowledge of the chip design, some of these dependencies can be understood. However, most others are unknown. We wish to develop an algorithm that will accurately predict chip quality based on a given set of measurements and in turn provide the quality of the wafer. This will allow early classification of failing chips that normally wouldn't be caught until much later in the testing process. This would account for significant savings in test time on poor quality wafers, as well as provide early feedback to the fabrication plant on the quality of their process.

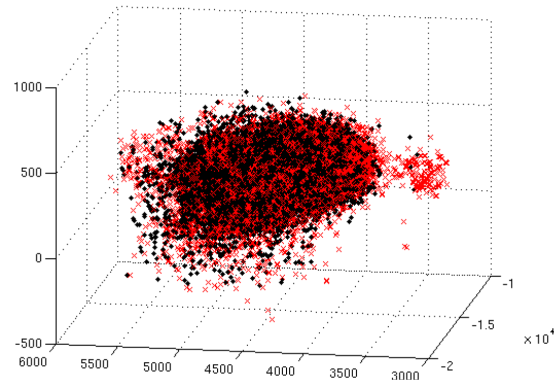


Fig. 1. Principal Component Analysis (PCA) Reduction of The Data Space to a 3-dimensional View

II. Motivation

The goal of the model is to create a method by which wafer quality prediction can be made and fed back to the fab. Since these structures are measured so early in the testing process and require very little overhead they are ideal for fab feedback. With this added classification mode, wafers with high defect rate can be isolated and understood using the more complex, and slower to measure, parametric structures. Placing this model in-line during a device bring-up can help to isolate issues in the fab process and improve yield, and in mature processes act as an indicator of fab issues.

In some cases defects aren't detected until well after chips have been taken off the wafer and are being tested in system. At this point the process of removing the chip and isolating the fail is very costly. With the added layer of a learning algorithm selecting out wafers, the number of defects in systems should reduce.

III. Background

Roughly 70,000 chips were selected for this study. Their respective VCO data and defect classifications were collected.

The process selected contains five groupings of ten values, each at various locations on the chip. The goal is wafer identification, but quality is measured by chip. So individual chips are separated out in the data sets we collect.

In figure 1 we show a subset of the data set. Each is normally distributed. Some outliers exist, but they are already easily identified, and most represent issues with the measurement system, which are filtered out in advance.

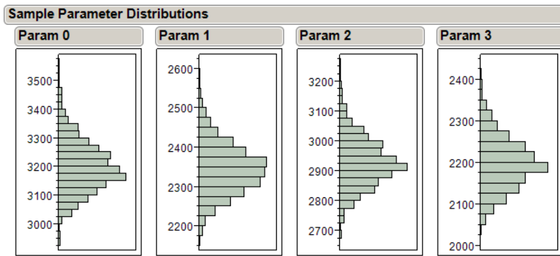


Fig. 2. Sample Parameter Distributions.

A. Definitions

For the purpose of this paper we define a feature vector $x^{(i)} \in \mathbb{R}^{50}$ denoting a specific chip and a corresponding value of $y^{(i)} \in \{-1, 1\}$ denoting quality of the chip, where 1 means the chip is determined to be defective. We define output $y'^{(i)} \in \{-1, 1\}$ as the predicted label of the sample $x^{(i)}$. Additionally, we define per chip wafer assignment as a column vector w, where $w^{(i)} \in \{1, \ldots, N\}$ is the wafer number for chip i.

IV. Process Overview

The overall process of developing and applying the learning algorithm can be represented in a series of successive steps that normalize and prepare the data, find an appropriate training set, and fit the data. Figure 3 represents this process schematically and subsequent sections provide more insight into details of each step of the process.

The final outcome of the process is a model suitable for detecting "low-yield" wafers, where the yield threshold τ can be varied at the expense of accuracy.

Fig. 3. Data Modeling Process.

V. Data Filtering

TEST measurement errors cause some VCO registers to report back erroneous values. In some cases the patterns fail to initialize the test correctly, which results in a register value of 0. In other cases the register fails to clear, resulting in a value well outside of the expected range. Since our data set is so large (70,000+ test cases) and failing measurements do not provide any information about the quality of the chips themselves, we removed chips with values of any individual feature being equal to 0 or outside the range of $\mu \pm 4\sigma$ of the distribution of measured values.

$$X := \{x \in X : i = 1 \ldots m, x_i \neq 0, |x_i - \mu_i| \leq 4\sigma_i\}$$

Some wafers had systemic problems that required no additional screening. In these cases some fab process caused all but a handful of parts to fail gross functional testing, continuity or power shorts, and had no VCO values to report. In these cases the remaining chips, usually less than 25% of the wafer, all failed. Since these skew our per wafer classification and have no added value (they are easily identified as process issues) they were removed from out base set.

VI. Scaling

SEVERAL data scaling strategies were tried to achieve optimal results in chips quality classification. Based on prior knowledge, taking the proportion of parameters against each other was attempted. This yielded minor improvements in linear and logistic regression models but provided little improvement when using a support vector machine.

Various VCO parameters represent different types of measurements, such as trace length or delay in gate response, and have widely varying ranges. Scaling parameters using a unified scheme across all dimensions therfore results in reduced accuracy. Instead we scale each feature individually. We found that scaling values to the range $[0, 2]$ worked to optimize performance while keeping the range wide enough so that rounding errors did not affect our model. Other ranges where considered but all performed worse or the same. A linear scaling method from the original values to a set [0,2] inclusive was chosen:

$$x_j^{(i)} = 2\frac{x_j^{(i)} - \min_j(x_j)}{\max_j(x_j) - \min_j(x_j)}$$

VII. Support Vector Machine Utilization

Various machine learning algorithms were attempted to model the data, including linear regression, logistic regression, and support vector machines. Utilizing principal component analysis, the dimensionality of the problem was reduced to 3 and 2 dimensions, and the visual representation of this data set (see figure 1, figure 4) provided compelling evidence that the data under consideration was non-linearly separable even in fairly high dimensional spaces. Therefore, SVM with non-linear Kernels tended to perform better and was chosen as a modeling algorithm for this application.

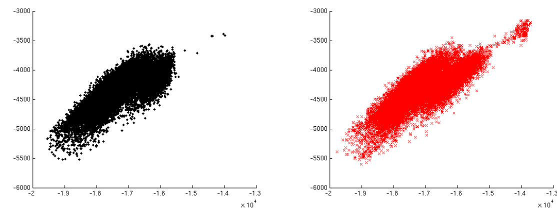


Fig. 4. 2D Representation Of Chip Data. Red Represent Failing Chips.

A. Kernel Selection

A number of kernel functions were attempted to model the data with following conclusions:

- Linear Kernel suffered from large bias and tended to underfit the data regardless of the training set size, causing both training set error and generalization error to be large

- Polynomial Kernel suffered from large variance, which caused it to overfit the training set data and perform very poorly (worse than guessing) on a larger test data set
- Radial family of kernel functions, such as Gaussian, Exponential, Laplacian, tended to perform better overall due to the nature of the data. The Gaussian RBF Kernel was chosen because it provided an optimal balance of sensitivity to function parameters and runtime performance

$$K(x^{(i)}, x^{(j)}) = exp(-\gamma||x^{(i)} - x^{(j)}||^2)$$

The Gaussian Kernel maps the input space to an infinite-dimensional feature space on which the SVM algorithm is used to perform a search for a separating hyperplane [3]. This allows us to deal with the non-linearly separable data in the original feature space. The tightness of the fit of the hyperplane is controlled by the parameter γ as well as the standard SVM mechanism C. Then, the classification function becomes:

$$f(x) = sgn(\sum_{i=1}^{n} y_i \alpha_i \exp(-\gamma||x - x_i||^2) + b)$$

B. Parameter Selection

To deal with the non-linearly separable data we introduce a cost factor C that allows the algorithm to deliberately mis classify examples while paying a premium for doing so. This allows the maximization of the margin while at the same time ensuring that as many examples as possible are classified correctly.

In most cases the test data will be disproportionately split between "good" samples and "bad" samples, with "bad" samples accounting for roughly 35-45% of the data. The initial assumption was that utilizing a split cost factor would be beneficial in improving accuracy of our overall algorithm. We defined C_+, C_-, such that our original SVM problem was modified:

$$\min_{y,w,b} \frac{1}{2}||w||^2 + C_+ \sum_{i}^{n+} \zeta_i + C_- \sum_{i}^{n-} \zeta_i$$

so that the positive and negative examples are separated for classification. We saw an increase in precision/recall values of the algorithm both on the training set (via cross-validation) and the test set. However, overall accuracy of the algorithm suffered, negating any improvements that may have been achieved.

An iterative approach was used for selecting the best values of parameters γ and C. Figure 5 shows a plot of achieved accuracy as a function of the SVM parameters.

A smaller training set of 2000 data points, with k-fold cross validation utilizing 5 disjoint subsets, was used to determine projected algorithm accuracy. The maximum accuracy was achieved at the values shown in figure 6.

C. Feature Selection

Our original data set includes readings from 50 different VCOs on the chip, which implies that each element of training and test set is a vector in $\mathbb{R}^{50}$. Two approaches to reducing the dimensionality of the problem were considered for the purposes of speeding up the computational runtime as well as improving overall accuracy of the algorithm. We know that the input vectors actually represent data about various subsections of a chip, with distinct sub-groupings of 10 elements repeated 5 times, such that elements can be broken up into the following set of sets:

$$\begin{aligned} X_g = \{ & \{x_1 \dots x_{10}\} \\ & \{x_{11} \dots x_{20}\} \\ & \{x_{21} \dots x_{30}\} \\ & \{x_{31} \dots x_{40}\} \\ & \{x_{41} \dots x_{50}\}\} \end{aligned}$$

Since the groupings of elements describe specific areas of the chip in a repetitive manner, it seemed reasonable to utilize principal component analysis to reduce the dimensionality of the problem to 5 dimensions (corresponding to each area of the chip). The PCA reduction was performed by first finding a unit length vector to satisfy the following condition for each subset:

$$u_{sn} = \arg\max_{\vec{u}} u^T \left(\frac{1}{m} \sum_{i=1}^{m} x_{sn}^{(i)} {x_{sn}^{(j)}}^T \right) u$$

where x_{sn} is a 10-dimensional subset of each element x starting at element index s. Then the reduction was repeatedly performed on each subset:

$$x'_i = < u_{sn}, x_{sn} >$$

The resulting 5-dimensional subset was used in the same SVM training strategy as described earlier. The expectation was that using PCA in such a way would naturally decrease the amount of noise generated by discrepancies in measurements across chip subsections and improve accuracy as well as improve runtime by reducing the size of the classification problem. However, in practice, reducing dimensionality in such a way actually hurt overall accuracy of the algorithm, leading us to believe that discrepancies between measurements within one subsection of a chip are indicative of the overall quality of the chip.

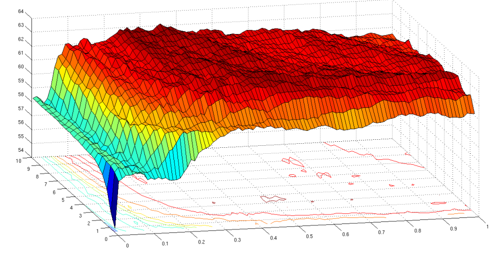


Fig. 5. Algorithm Accuracy As A Function of Parameters γ, C

γ	C	Reported Accuracy
0.3548	10	63.47%

Fig. 6. Maximizing Parameters For Gaussian Kernel SVM Algorithm

Having determined the independence of data points within physical chip groups we have attempted to perform independent feature selection via a forward search algorithm. A k-fold cross validation on a sample size of 2,000 chips was used with optimal parameters found earlier in the process to select a subset of features that would eliminate noise and provide better accuracy. The parameters were also varied slightly for a number of independent runs of f-search algorithm to account for the potential need for adjustment. We have discovered that although the nature of the features is repetitive (each 10th VCO repeats the same measurement on a different subsection of the chip) there is no optimal subset of features F' smaller than the original set that produces better accuracy.

$$\arg\max_{F' \subseteq F} \left(\sum_{i=1}^{m} \mathbf{1}\{\text{SVM}(x_{F'})^{(i)} = y^{(i)}\} \right) = F$$

where $x_{F'}$ denotes an input sample x with only features F' selected and SVM(x) denotes a prediction made by our algorithm.

D. Training Set Size Selection

Our final consideration for the model is the optimal size of the training set. In the context of this study the impact of the training set size on algorithm performance is a very important consideration because it determines how soon into production of the new generation of chips we can achieve acceptable accuracy on waver yield predictions. Our intuition is that the training error and generalization error should converge at some optimal training set size. To verify our assumption we can iteratively determine training error and generalization error for various sizes of training sets using the optimal parameters and features. Figure 7 is a graphical representation of the training set and test set error as a function of the training set size.

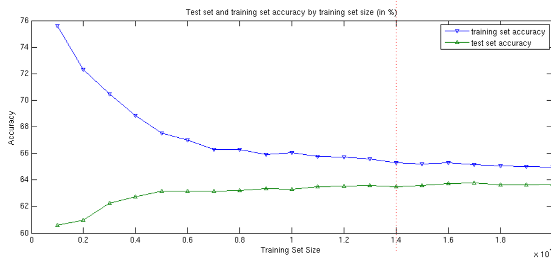
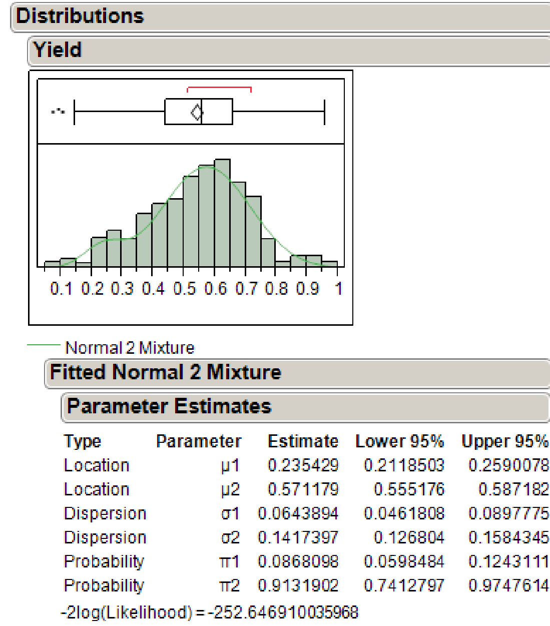


Fig. 7. Training Set and Test Set Error by Training Set Size

It is evident from the graph that increasing the size of the training set past ∼14,000 elements does not generate significant change in accuracy. It is clear that this training set size provides optimal balance between accuracy of the algorithm and the computational resources needed to run it.

Another important implication of the figure above is that there is little benefit to implementing this system using online learning, since large amounts of extra training data does not provide additional accuracy for algorithmic predictions.



Type	Parameter	Estimate	Lower 95%	Upper 95%
Location	μ1	0.235429	0.2118503	0.2590078
Location	μ2	0.571179	0.555176	0.587182
Dispersion	σ1	0.0643894	0.0461808	0.0897775
Dispersion	σ2	0.1417397	0.126804	0.1584345
Probability	π1	0.0868098	0.0598484	0.1243111
Probability	π2	0.9131902	0.7412797	0.9747614

-2log(Likelihood) = -252.646910035968

Fig. 8. Wafer Yield Distrobution

VIII. Per Wafer Classification

OUR original goal was a per wafer classification model to identify wafers to be tested with the more intensive screening process. With this in mind we now classify the chips by wafer and look for yields that fall below some set threshold τ. Looking at the distribution of yield per wafer figure 8, we see a bimodal nature in the distribution. The mean ($\mu 1$) of the lower distribution is the set were interested in identifying.

These are wafers where yield fell outside the normal distribution and beyond an acceptable production level. From this we can chose a boundary point for τ to be $\mu 1 + \sigma \approx 0.38$.

Defining an equation for wafer yield on of the test set $W_r^{(j)}$ and wafer yield as predicted from our SVM model $W_p^{(j)}$,

$$W_r^{(j)} = 1\left\{ \frac{\sum_{y:i=1}^{K} 1\{y^{(i)} = -1 \wedge w^{(i)} = j\}}{\sum_{y:i=1}^{K} 1\{w^{(i)} = j\}} > \tau \right\}$$

$$W_p^{(j)} = 1\left\{ \frac{\sum_{y:i=1}^{K} 1\{y'^{(i)} = -1 \wedge w^{(i)} = j\}}{\sum_{y:i=1}^{K} 1\{w^{(i)} = j\}} > \tau \right\}$$

We can then define an equation for accuracy of our model by wafer as,

$$Accuracy = \frac{1}{N} \sum_{w:j=1}^{N} 1\left\{ W_r^{(j)} = W_p^{(j)} \right\}$$

Using this a plot of the accuracy as the split point classification τ is changed can be generated, figure 9. Which shows our best accuracy with the highest split point to be ~ 0.38. This matches out desired and expected split point based on the bimodal nature of the data.

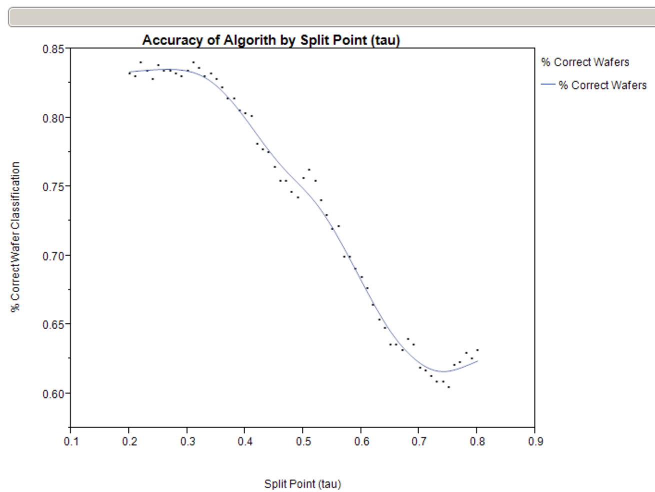


Fig. 9. Accuracy by The Split Point τ.

IX. Conclusions

On a per chip basis our model does not perform well enough to be effectively used in production. This result is not surprising, considering previous attempts to use the data to identify individual defective chips by searching for distribution based indicators. However, using the SVM model in conjunction with wafer level classification, the accuracy of prediction of low-yield wafers can be as high as 81%. Additionally we can develop a method of scaling projected yields at the expense of accuracy, so higher yield wafers can be identified as well. Given this algorithm, wafers with high rate of defects can be isolated and screened for fabrication defects resulting from machine calibration issues. This level of early classification assists in identification of process defects and possible machinery issues, which in the past could only be identified after additional screening, which results in cost savings of several days of production per identified wafer.

References

[1] Boser, B.E.; Guyon, I.M.; and Vapnik, V. 1992 A training algorithm for otimal margin classifiers. *Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, ACM 144-152

[2] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM : a library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

[3] Scholkopf B., Sung K., Burges C., Girosi F., Niyogi P., Poggio T., Vapnik V., *Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers*, MIT, 1996

[4] T. Joachims, *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*, B. Schlkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[5] K. Morik, P. Brockhausen, and T. Joachims,*Combining statistical learning with a knowledge-based approach - A case study in intensive care monitoring*. Proc. 16th Int'l Conf. on Machine Learning (ICML-99), 1999.