UNIVERSITY OF CALIFORNIA

SANTA BARBARA

Data Analytics in Test:

Recognizing and Reducing Subjectivity

A dissertation submitted in partial satisfaction

of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Sebastian Siatkowski

Committee in charge:

Professor Li-C. Wang, Chair

Professor Margaret Marek-Sadowska

Professor Forrest Brewer

Professor Yuan Xie

June 2017

This dissertation of Sebastian Siatkowski is approved.

Professor Margaret Marek-Sadowska

Professor Forrest Brewer

Professor Yuan Xie

Professor Li-C. Wang, Committee Chair

June 2017

Data Analytics in Test:
Recognizing and Reducing Subjectivity

Sebastian Siatkowski

Dedicated to my mother Wanda,
my father Leszek,
and my sister Patrycja.

Acknowledgements

First and foremost, I would like to express sincere gratitude and appreciation to my advisor, Professor Li-C. Wang, for his guidance, mentorship, and friendship over the past five years. Li generously provided opportunities, advice, and support that helped me mature not only as a more well-rounded, innovative engineer, but also as a more aware human being. The time I spent working for and alongside Li has allowed me to pick up just a fraction of his incredible ability to break down any complex problem into its fundamental components, but with that alone I feel well equipped to take on any challenges that I may encounter. I will cherish the relationship we developed for the rest of my life.

I acknowledge everyone in the research community who has helped set the foundation for my research. In particular, I would like to thank Professor Wang's former student and my mentor throughout my internships at Freescale and NXP, Nik Sumikawa. Nik's prominent contributions to the field allowed me to immediately dive into solving problems that made an impact. I am grateful for his invaluable guidance throughout my research career and his friendship. I also want to thank Magdy Abadir and LeRoy Winemberg who enabled the research collaboration with Freescale and provided access to production data.

I graciously thank the current students in Professor Li-C. Wang's group, Jay Shan, Matt Nero, Kuo-Kai Hsieh, and Ahmed Wahba, for their friendship and for making spending countless hours cooped up in lab enjoyable. Being surrounded by such intelligent individuals who were always available for engaging research discussions was vital to my success as a researcher. In addition, I want to thank my close friends Arthur Koner, Luigi Mangiacotti, Kevin Pusateri, and Ali Abbasinasab, who were there when I needed to take my mind off research, which kept me sane over the years.

I appreciate everyone who has provided me with support throughout my studies.

I would like to express my gratitude to all the ECE faculty at UCSB, especially my committee members Professor Margaret Marek-Sadowska, Professor Forrest Brewer, and Professor Yuan Xie. I am honored to have been given the opportunity to learn from such brilliant and inspirational educators. I would also like to recognize Dr. Amr Haggag who was a member of my committee for a portion of my graduate journey for his valuable advice. In addition, I am very thankful to Mrs. Bren Simon for her generosity throughout my entire college education. It is hard to imagine how I would have gotten this far without her financial support.

Most of all, I want to thank my family for always being there for me. My mother Wanda's nurture was the single most influential factor in shaping the person I am today. I am forever grateful for the sacrifices she had to make to provide me with the comfort and the opportunities I have had in life. I appreciate my father Leszek for the hardships he had to deal with to allow me to take the path I took. And, I cannot imagine a more supportive sibling than my sister Patrycja, who has been my best friend for as long as I can remember. I love you all.

Curriculum Vitae

Sebastian Siatkowski

EDUCATION

2013 – 2017	PhD in Electrical and Computer Engineering, University of California, Santa Barbara.
2012 – 2014	M.S. in Electrical and Computer Engineering, University of California, Santa Barbara.
2008 – 2012	B.S. in Computer Engineering, University of California, Santa Barbara.

PUBLICATIONS

1. Li-C. Wang, Sebastian Siatkowski, Chuanhe (Jay) Shan, Matthew Nero, Nikolas Sumikawa, LeRoy Winemberg, *Some Considerations on Choosing An Outlier Method for Automotive Product Lines,* submitted to ITC, Oct. 2017

2. Sebastian Siatkowski, Li-C. Wang, Nik Sumikawa, LeRoy Winemberg, *Learning the Process for Correlation Analysis,* VTS, Apr. 2017

3. Kuo-Kai Hsieh, Sebastian Siatkowski, Li-C. Wang, Wen Chen, Jayanta Bhadra, *Feature Extraction from Design Documents to Enable Rule Learning for Improving Assertion Coverage,* ASP-DAC, Jan. 2017

4. Sebastian Siatkowski, Chuanhe (Jay) Shan, Li-C. Wang, Nik Sumikawa, W. Robert Daasch, John M. Carulli Jr., *Consistency in Wafer Based Outlier Screening,* VTS, Apr. 2016. Best paper recipient

5. Sebastian Siatkowski, Chia-Ling Chang, Li-C. Wang, Nik Sumikawa, LeRoy Winemberg, W. Robert Daasch, *Generalization of an Outlier Model into a "Global" Perspective,* ITC, Oct. 2015

6. Jeff Tikkanen, Sebastian Siatkowski, Nik Sumikawa, Li-C. Wang, Magdy S. Abadir, *Yield Optimization Using Advanced Statistical Correlation Methods,* ITC, Oct. 2014. Best paper recipient

FIELD OF STUDY

Electrical and Computer Engineering — Professor Li-C. Wang

Abstract

Data Analytics in Test:

Recognizing and Reducing Subjectivity

by Sebastian Siatkowski

Applying data analytics in production test has become a widely adopted industrial practice in recent years. As the complexity of semiconductor devices scales and the amounts of available test data continue to grow, the research direction in this field is forced to shift away from solving specific problems with ad hoc approaches and demands for deeper understanding of the fundamental issues. Two data-driven test applications where this shift is apparent are production yield optimization and defect screening, where the respective underlying data analytics approaches are correlation analysis and outlier analysis. A core issue present in these two approaches stems from the subjectivity that is inherent to data analytics. This dissertation delves into how subjectivity manifests itself and what can be done to reduce it with respect to the two test applications.

Outlier analysis is an approach used for identifying anomalies. The main goal of outlier analysis in test is to capture statistically outlying parts with the hope that their abnormal behavior is attributed to some defectivity. During creation of an outlier model, the decisions about outlying behavior in the existing data are made by utilizing known failures and the test engineer's best judgment. In practice, outlier screening methods are simply used for transforming data into an outlier score space. Even if outlier analysis techniques are able to successfully classify a dataset into inliers and outliers, outlier models require thresholds to be decided. A concept called Consistency is introduced to provide an objective data-driven way to evaluate outlier models by

utilizing all available data. The key observation underlying this concept is that outlier analysis should be immune to noise introduced by sources of systematic variation.

Correlation analysis is a process comprising a search for related variables. The application of production yield optimization involves searching for correlation between the yield and various controllable parameters. The goal of this process is to uncover parameters that, when adjusted, can result in yield improvement. This analytics process is subjective to the perspective of the analyst and the quality of the result is highly dependent on the analyst's previous experiences. In order to reduce the subjectivity in this application, a process mining methodology is introduced to learn from the experiences of analysts. The key advantage of this methodology is that in addition to having the capability to record and reproduce these analyses, it can also generalize to analytics processes not contained in the learned experiences.

Contents

Curriculum Vitae **vii**

Abstract **viii**

List of Figures **xv**

List of Tables **xix**

List of Abbreviations **xx**

1 Introduction **1**

1.1 **Data Analytics in Test** 1

1.1.1 Correlation Analysis 3

1.1.2 Outlier Analysis 4

1.2 **The Added Benefit of Data Analytics in Test** 7

1.3 **Related Works and Approaches** 8

1.4 **The Issue of Subjectivity** 10

1.4.1 Subjectivity in Correlation Analysis 11

1.4.2 Subjectivity in Outlier Analysis 13

1.5 **Dissertation Organization** 15

2 Yield Optimization Using Advanced Statistical Correlation Methods **16**

2.1 **Overview** 16

2.2 **Introduction** . . . 17
2.3 **Potential issues with the intuitive methodology** . . . 22
2.3.1 Need for multivariate analysis . . . 22
2.4 **Multivariate correlation and statistical dependence** . . . 25
2.4.1 Canonical Correlation Analysis (CCA) . . . 27
2.4.2 Analysis of test A in bin 26 . . . 29
2.4.3 X_1, X_2, X_3 types of fails (removing X_4 fails) . . . 31
2.4.4 Analysis of test D (Bin 25) . . . 32
2.4.5 Summary of the first finding - parameter PP1 . . . 33
2.4.6 Note on applying CCA in location-based analysis . . . 34
2.5 **The subset discovery problem** . . . 35
2.5.1 Assumption for subset discovery to be useful . . . 36
2.5.2 Heuristic to approach the problem . . . 37
2.5.3 Analysis of X_1-X_3 types of fails from test A . . . 38
2.5.4 Result illustration . . . 39
2.5.5 Double check X_4 types of fails from test A . . . 40
2.5.6 Summary of findings . . . 41
2.6 **Risk evaluation** . . . 42
2.6.1 Kernel CCA (KCCA) looks for non-linear correlations . . . 42
2.6.2 Kernel CCA as a statistical independence test . . . 44
2.6.3 Practical implementation of kernel CCA . . . 45
2.7 **Yield improvement based on silicon results** . . . 48
2.8 **Summary** . . . 49

3 Learning the Process for Correlation Analysis **50**
3.1 **Overview** . . . 50
3.2 **Introduction** . . . 50

3.3 **Perspectives in yield optimization** . . . 53
3.3.1 What contributed to the success in Chapter 2 . . . 54
3.4 **The Learning Problem** . . . 55
3.4.1 Unsuccessful analytics trials . . . 57
Unsuccessful example 1 . . . 57
Unsuccessful example 2 . . . 58
3.5 **Learning the perspectives** . . . 59
3.6 **Designing The Process Steps** . . . 63
3.7 **Applying PM Model** . . . 67
3.7.1 Learning a PM model . . . 68
3.8 **Limitations of the PM Model** . . . 72
3.9 **Summary** . . . 73

4 Generalization of an Outlier Model into a "Global" Perspective 74
4.1 **Overview** . . . 74
4.2 **Introduction** . . . 74
4.2.1 Multivariate outlier example . . . 77
4.2.2 Temporal and spatial uncertainties . . . 78
4.3 **Understanding the uncertainties** . . . 79
4.3.1 Further illustration of temporal uncertainty . . . 80
4.3.2 Further illustration of spatial uncertainty . . . 81
4.3.3 Analyzing the result in Figure 4.1 . . . 82
4.4 **Identifying "gross" outliers** . . . 84
4.4.1 The concept of marginality test . . . 84
4.4.2 Using the N most similar wafers . . . 84
4.4.3 Examples of marginality test . . . 85
4.4.4 Gross outliers in view of Figure 4.1 . . . 87

4.4.5 Gross outliers vs. marginal outliers . . . 88
4.5 **The proposed outlier analysis approach** . . . 89
4.5.1 A "big data" perspective . . . 89
4.5.2 Issue with using a DPAT or AEC model . . . 90
4.5.3 Adaptive k value and its potential issue . . . 91
4.6 **Probability-based outlier evaluation** . . . 92
4.6.1 Estimating probability of occurrence . . . 92
4.6.2 Heuristic for fast probability estimate . . . 93
4.6.3 Probability-based marginality test . . . 94
4.6.4 Evaluating multiple potential outliers . . . 94
4.6.5 Deciding potential outliers . . . 95
4.7 **Probability-based online outlier evaluation** . . . 95
4.7.1 Handling the first b wafers . . . 96
4.7.2 Online outlier vs. Global outlier . . . 97
4.7.3 Comparison to earlier results . . . 97
4.8 **Comprehensive experimental results** . . . 100
4.9 **Summary** . . . 106

5 Consistency in Wafer Based Outlier Screening **108**
5.1 **Overview** . . . 108
5.2 **Introduction** . . . 108
5.3 **Potential inconsistency among methods** . . . 111
5.3.1 Two test examples . . . 112
5.4 **Consistency check** . . . 115
5.4.1 Finding minimum consistent threshold . . . 117
LA . . . 119
SPAT . . . 120

Noise band 120
5.5 **Detecting systematic shift** 122
5.5.1 Impact of clustering on consistency check 124
5.5.2 Finding no DPAT-consistent outlier 125
5.6 **Summary** 126

6 Conclusion and Future Work **127**
6.1 **Conclusion** 127
6.1.1 Subjectivity Reduction in Correlation Analysis 128
6.1.2 Subjectivity Reduction in Outlier Analysis 130
6.2 **Future Research Directions** 131
6.2.1 Learning the Process of Outlier Analysis 132
6.2.2 Applicability of Outlier Methods 133

Bibliography **135**

List of Figures

1.1 Data typically available in the production flow that is of interest to test data analytics . . . 1
1.2 Three basic components in outlier analysis . . . 5
1.3 High Level Overview of Analytics Process . . . 11

2.1 Illustration of yield fluctuation and our goal . . . 17
2.2 Bins of fails and their fluctuations . . . 19
2.3 Failing statistics based on individual tests . . . 19
2.4 Measurement sites and a fluctuation example . . . 20
2.5 Illustrating the starting point of this work . . . 21
2.6 Correlation can be sensitive to outliers . . . 22
2.7 Examples of site-site pairwise correlations . . . 23
2.8 Discrete test A and continuous test D . . . 24
2.9 Examples of failing wafer heat map . . . 25
2.10 Data Matrices . . . 28
2.11 CCA results for Test A . . . 30
2.12 Further illustration of results shown in Figure 2.11 . . . 31
2.13 CCA with new random vector $X = (X_1, X_2, X_3)$ . . . 32
2.14 Encoding a distribution into a multivariate vector . . . 32
2.15 Canonical Correlation vs. Pearson Correlation . . . 33
2.16 PP1 is correlated to the mean and variance of bin 25 . . . 34

2.17 Partitioning X_4 type of fails based on their locations 35
2.18 Subset discovery found two process parameters, PP2 and PP3, highly correlated to X_1-X_3 types of fails in bin 26 39
2.19 Subset discovery found two more process parameters, PP4 and PP5, correlated to X_1-X_3 types of fails in bin 26 40
2.20 Subset discovery confirms X_4 type of fails highly correlated to PP1 while X_1-X_3 types of fails do not 41
2.21 Illustration of kernel CCA 43
2.22 Kernel CCA risk evaluation on known results 46
2.23 Risk evaluation with respect to adjusting parameter PP1 46
2.24 Detailed analysis of the test in bin 31 vs. PP1 47
2.25 Silicon split-lot results show yield improvement 48

3.1 Analytics can be viewed as an iterative search process 51
3.2 Examples of high correlations found 55
3.3 An example of uncovering the temporal effect 56
3.4 One example triggering unsuccessful search 58
3.5 Another example triggering unsuccessful search 58
3.6 Simple state merging example 60
3.7 Another state merging example 61
3.8 Dimensions to consider in designing process steps 64
3.9 Yield issue due to cold/hot voltage tests 68
3.10 2-prefix PM model learned from the 39 traces 70
3.11 Finding association(left); Finding statistical correlation (right) 71
3.12 Association by finding two lines 72

4.1 Comparing three univariate outlier methods 76
4.2 Comparing two multivariate outlier methods 77

4.3 Temporal uncertainty across 756 tests . . . 80
4.4 Spatial uncertainty across 756 tests . . . 81
4.5 Temporal fluctuation and its impact to DPAT limits . . . 82
4.6 AEC outliers, and comparing to DPAT outliers . . . 83
4.7 Illustration of 300 most similar wafers to the wafer containing a marginal outlier . . . 85
4.8 Examples of marginal and gross outliers . . . 86
4.9 Gross outliers in view of the Venn diagram shown in Figure 4.1. A(B): A is the number of gross outliers and B is the number of outliers copied from Figure 4.1. . . . 87
4.10 Marginal vs. gross vs. shared gross outliers . . . 88
4.11 Outlier analysis following a big data perspective . . . 90
4.12 Potential outliers decided by a DPAT or AEC model . . . 90
4.13 Under-screen example for PPM target $= 10$ . . . 91
4.14 Issue with adaptive k value (10 PPM target) . . . 92
4.15 Illustration of kernel density estimation (KDE) . . . 93
4.16 Deciding potential outliers . . . 95
4.17 Setup for online outlier evaluation . . . 96
4.18 Comparison to result in Figure 4.10-(b) . . . 98
4.19 Comparison to result in Figure 4.13-(a) . . . 99
4.20 Result summary for over-screen cases . . . 100
4.21 Interesting over-screen cases . . . 102
4.22 Result summary for under-screen cases . . . 103
4.23 Interesting under-screen cases . . . 104

5.1 Test example 1 . . . 113
5.2 Test example 2 . . . 113

5.3 Min/Max value plot for test example 1 114
5.4 Min/Max value plot for test example 2 115
5.5 Illustration of the setting for consistency 116
5.6 % of shared outliers by all three methods 117
5.7 Results with minimum consistent threshold - DPAT 118
5.8 Illustration of test index 151 result 119
5.9 # of SPAT-consistent outliers (vs DPAT) 121
5.10 SPAT noise bands vs. DPAT noise bands 122
5.11 Two examples to illustrate clustering results 124
5.12 Impact of clustering on consistency check 124
5.13 An example of no DPAT-consistent outlier 125

List of Tables

2.1 Subset canonical correlations for four parameters PP2-PP5 found to have high correlations to the X_1-X_3 types of fails 38

2.2 Confirming strong correlation between X_4 fails and PP1 41

2.3 Summary of findings and supporting evidences 42

3.1 Traces used for producing figures 66

3.2 Prefix length vs number of traces 69

4.1 Actual PPM for a PPM target across all tests 81

4.2 Result summary on two automotive product lines 105

5.1 DPAT and LA disagree on consistent outliers 119

List of Abbreviations

AA	**A**ssociation **A**nalysis
AEC	**A**utomotive **E**lectronics **C**ouncil (DPAT)
AMS	**A**verage **M**odel **S**tability
CC	**C**anonical **C**orrelation
CCA	**C**anonical **C**orrelation **A**nalysis
DPAT	**D**ynamic **P**art **A**verage **T**esting
E-test	**E**lectrical **Test** (process parameter measurement)
ECID	**E**lectronic **C**hip **Id**entification
GDBC	**G**ood **D**ie in a **B**ad **C**luster
HA	**H**eatmap **A**ssociation
IC	**I**ntegrated **C**ircuit
KCCA	**K**ernel **C**anonical **C**orrelation **A**nalysis
KDE	**K**ernel **D**ensity **E**stimation
KPCA	**K**ernel **P**rincipal **C**omponent **A**nalysis
KS	**K**olmogorov-**S**mirnov
LA	**L**ocation **A**veraging
LR	**L**inear **R**egression
Mah	**Mah**alanobis (distance)
NNR	**N**earest **N**eighbor **R**esidual
PAT	**P**art **A**verage **T**esting
PC	**P**rincipal **C**omponent
PCA	**P**rincipal **C**omponent **A**nalysis
PM	**P**rocess **M**ining
PP	**P**rocess **P**arameter
PPM	**P**arts **P**er **M**illion
RDPAT	**R**obust **D**ynamic **P**art **A**verage **T**esting
RF	**R**adio **F**requency
SC	**S**tatistical **C**orrelation
SCC	**S**ubset **C**anonical **C**orrelation
SoC	**S**ystem **o**n a **C**hip
SPAT	**S**tatic **P**art **A**verage **T**esting
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

1.1 Data Analytics in Test

The semiconductor production process comprises many stages from design, to manufacturing, to test, to in-field deployment. At each step along the way, data is generated and collected, leading to parts typically having data for intended specifications, usage profile of process tools and chambers, process parameters (E-tests), wafer sort tests, burn-in, final tests, and in-field performance. These data stages are visualized in Figure 1.1. Availability of this tremendous amount of data has greatly influenced the interest in and profitability of applying modern data analytics in Test, which has presented many research opportunities [1].

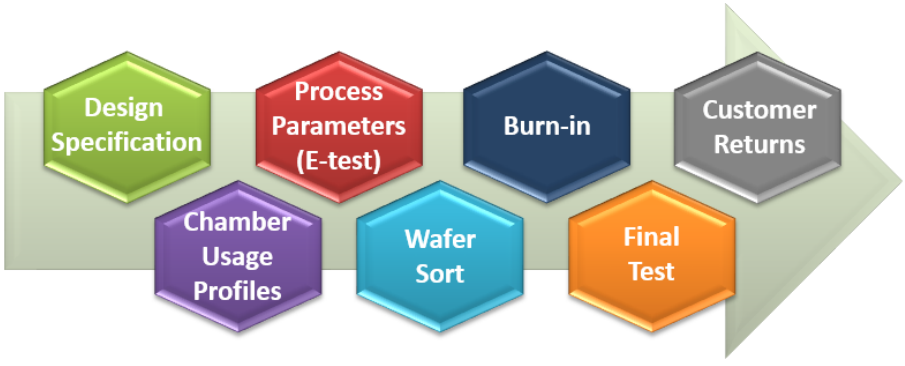


FIGURE 1.1: Data typically available in the production flow that is of interest to test data analytics

Research aimed at utilizing data mining along with statistical models in Test has been ongoing for over a decade [2][3]. A broad objective for data analytics is to learn from data. In view of this objective, data mining is seen as a *knowledge discovery* process [4]. In the case of Test, this knowledge contains information on how the test flow can be improved in terms of cost and quality.

The applications of data analytics in Test are quite extensive. Some of these applications include yield optimization [5][6], test cost reduction [7], customer return analysis [8][9], burn-in reduction [10][11], and outlier detection [12][13]. Furthermore, the tools used within these applications may differ when additional data aspects are taken into consideration, such as wafer patterns [14], spatial correlations [15], multivariate test spaces [16], data robustness [17], process variations [18], or systematic defects [19]. Though many of the application specific solutions are intricate and efficient, they often tend to solve ad hoc problems which may not be desirable from the research standpoint. The work in this dissertation aims to preserve the value of the multitude of research in this field by introducing novel approaches which combine multiple solutions and are easily expandable .

The ease of access to modern data mining tools may be one of the factors that contributed to data analytics finding so many applications in Test. Multiple software machine learning packages and libraries are available online, with many of them being free or open-source projects. One such library is Scikit-learn [20], available under the minimally imposing BSD (Berkeley Software Distribution) license, which was utilized to produce many of the results in this thesis.

Though numerous data analytics problems exist in Test, the underlying approaches for solving those problems have some fundamental commonalities. At the core of many yield optimization, test cost reduction, and some burn-in elimination problems lies *correlation analysis*. The problems of outlier screening, customer return analysis, and burn-in reduction frequently rely on *outlier analysis*. Sections 1.1.1 and 1.1.2 below

introduce the concepts of correlation analysis and outlier analysis, respectively, and discuss the nature of their applications in Test.

1.1.1 Correlation Analysis

A definition of *correlation* widely accepted by the statistics community is:

> *The degree to which two or more attributes or measurements on the same group of elements show a tendency to vary together.*

The measure of correlation between sets of variables can be calculated in multiple ways. For instance, Pearson correlation can be used to evaluate the linear relationship between variables, while Spearman correlation can be used to evaluate the monotonic relationship between variables. To avoid constraining correlation to any particular method, further discussions apply to any correlations satisfying the above definition.

In the context of Test, correlation is used to identify related measurements in data from within or across production stages. The purpose of finding correlation varies across applications.

Yield optimization techniques are often employed when the *yield* ($\frac{\#\ passing\ parts}{\#\ total\ parts}$) of a product is lower than expected. For these techniques, correlation serves as the metric that is used to identify controllable parameters which can be used to correct known sources of yield loss. Yield loss is typically manifested as failures at one of the test stages (i.e. wafer sort, burn-in, or final test). If correlation can be found between the failures and some E-test or manufacturing data, then some parameters can be identified as candidates for correcting the yield issue. Adjustment of those parameters can then lead to fewer parts exhibiting those failures, resulting in improved yield.

Test cost reduction, as the name suggests, aims to bring down the cost of production by cutting costs in the test stage. Through the use of data analytics this can be

done by identifying and removing redundant or unnecessary tests, or by implementing selective test methodologies. In either case, correlation is one metric that can be used to make such decisions. For example, when two tests are very highly correlated, an argument can be made for the case that running both tests is redundant. Finding more intricate correlations can allow for test reordering and partial testing methodologies where, for example, a part is only run on test C if the right conditions were met on its values for tests A and B.

Burn-in elimination can be fundamentally quite similar to test cost reduction. If a set of preceding tests can be shown to cover all the failures captured by burn-in, then the burn-in step can be removed. In practice this is difficult since burn-in aims to capture failures that are not identifiable through tests. That being known, instead of simply searching for correlation among existing tests, a common practice is adding high-voltage stress tests intended to mimic the early life acceleration performed by burn-in [21]. Then, correlation is used to justify burn-in elimination or devise selective burn-in strategies.

Correlation analysis is the core step of many data mining problems in Test. It is therefore not unreasonable to conjecture that studying and improving the process of correlation analysis could lead to a key contribution in the field. The work presented in Chapter 2 focuses on the application of correlation analysis within a yield optimization context and the work in Chapter 3 focuses on learning the process undertaken by a data analyst to resolve yield issues.

1.1.2 Outlier Analysis

Although outlier analysis is well researched, with many books covering the subject extensively, there is no single agreed upon definition of an outlier. One definition that has existed for a while and captures the essence of what an outlier is quite well was coined by Hawkins [22]:

> *An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.*

Outlier analysis is a popular approach for capturing parametric defects [17]. The goal of applying outlier analysis in Test is to capture defective parts that pass through the existing test flow. To achieve this goal, outlier analysis works under the assumption that it is possible that abnormal behavior within the test flow can be used to differentiate defective parts from good parts.

Outlier analysis is based on an outlier model. When developing such an outlier model, three components are to be considered, as illustrated by Figure 1.2. The first component is the set of samples to be analyzed together. This set is referred to as the *base set*. The second component is the method used to calculate an *outlier score* for each sample. This method is often called the outlier analysis method or just outlier method. After the outlier score calculation, samples are conceptually ordered by an *outlier rank*. Because the outlier score is intended to be comparable across base sets, the outlier rank can be established globally. The third component is a way to decide the *threshold* on the outlier rank. This threshold separates the samples into inliers and outliers.

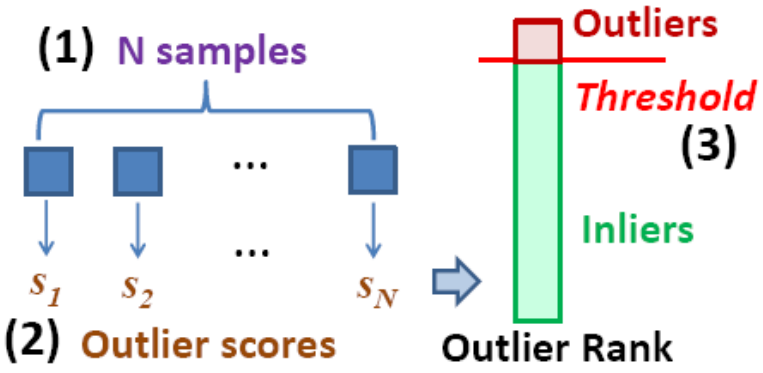


FIGURE 1.2: Three basic components in outlier analysis

Preemptive outlier screening is often adopted in the test flow for automotive products due to their extremely high quality requirement. In this application, an analyst

decides the threshold by looking at the data for some selected base set and outlier method. The objective is to improve the quality of the product by allowing for a yield reduction, with the hope that the screened outliers were bad parts or future failures. This approach follows the unsupervised leaning paradigm [23].

Customer return analysis is a reactive outlier analysis approach where a set of future failures is known. Utilizing this knowledge, an analyst can select a threshold that would have captured a failure, preventing similar parts from passing through the test flow in the future. The three components of the outlier model are often adjusted to minimize the effect that the model will have on yield loss. Though the same underlying outlier analysis components are used, this approach follows the supervised learning paradigm [23].

One aspect of burn-in reduction is quite similar to customer return analysis. The idea is that if parts failing burn-in could be captured in earlier test stages, then the need for burn-in could be reduced. From the analyst's standpoint, the experimental setup is almost the same. And, in either case, the goal is to discover future failures early in the production pipeline.

It is important to point out that outlier analysis alone often cannot justify making changes to the test flow. In practice, data analytics findings are presented to teams who posses the domain knowledge to assess if the findings make sense. For example, it may be necessary to check if a test found to screen a customer return is related to the failure mechanism.

Outlier analysis is used for multiple applications of data analytics in Test. The work in Chapters 4 and 5 of this dissertation focuses on exposing problems with current the outlier analysis approaches and exploring novel solutions.

1.2 The Added Benefit of Data Analytics in Test

One reason why data analytics is attractive to test applications is because simple analytics approaches do not require modifications to the existing test flows. A considerable portion of analytics can be applied in an offline fashion by using data that is already being collected.

The rising interest in applying data analytics in Test, both among academics and industry professionals, has largely been fueled by instances where data analytics was shown to add clear benefit to the existing processes. This section highlights some examples where such benefit was shown.

The use of outlier screening has been demonstrated to successfully capture customer returns [24]. Additionally, customer return parts have also been captured through multivariate outlier screening models at wafer sort [25].

In one case, successful test cost reduction for an RF/A IC was achieved through multivariate parametric testset optimization [7]. In another case, successful test cost reduction for an RF radio transceiver was accomplished through chip performance prediction [26].

One case of successful burn-in avoidance was demonstrated with the use of outlier analysis [27]. A case where burn-in reduction is accomplished showed how failures requiring longer burn-in times could be identified through analysis of parametric test results [11].

Some other problems in Test have also found successful data analytics applications. Analytics can be utilized to provide accurate yield estimates across design generations and among facility locations [28]. Data-driven solutions can be devised for identifying systematic faults in volume diagnosis [29]. In another problem, the cause for mismatch between decoupled plasma nitridation chambers was identified by a data analytics algorithm [30].

The impact of data analytics on semiconductor manufacturing has been held in high regard by industry leaders. A white paper from Intel [31] stated that big data was instrumental in manufacturing at smart factories, which led to improved productivity and quality. A white paper from Oracle [32] stated more broadly that big data solutions can help improve manufacturing efficiency and can lead to improved and timely decision making.

The promising impact of data analytics in Test has recently led to the formation of multiple companies wishing to capitalize on the advances in this field. These companies package analytics tools that simplify the task of applying data analytics for semiconductor manufacturers. The tools have already found successes, for example as shown by OptimalPlus [33] and by PDF Solutions [34].

Chapter 2 of this dissertation presents an example where correlation analysis leads to successful yield optimization. This example is supported by a silicon experiment, details of which are described within the chapter.

1.3 Related Works and Approaches

This section references works representing the industrial practice approaches as well as the state-of-the-art research solutions to the respective problems of yield optimization and outlier screening.

Yield is a subject that has been studied widely and extensively. For production yield improvement, existing efforts may include approaches based on yield modeling, volume diagnosis, and/or root-causing.

Earlier work such tried to identify the top parametric parameters that were most sensitive to yield and model their impact using multivariate regression [5]. Another work [18] used K-Means to cluster wafers into two groups, one with good yield and

one with poor yield. Kruskall-Wallis and decision trees were applied to identify process parameters that were most likely to explain the discrepancy between the two groups.

Volume diagnosis is an effective approach for yield improvement [6][35][36]. For example, one work [35] applied a novel statistical learning algorithm to produce accurate feature failure probabilities to better understand yield limiters. Another work [36] incorporated logic diagnosis data along with information on physical features in the layout to identify dominant defect mechanisms among failing dies.

For lithographic induced systematic issues, one work [19] proposed methods to extract features and cluster layout snippets to identify possible defect hotspots. These methods wish to identify systematic defects as yield limiters.

The focus in Chapter 2 of this work is specific to yield optimization in the mass production stage where large quantities of test data are available. The data comes mostly from parametric tests where root causing the failures could be difficult. Hence, the correlation approach is applied as an alternative to the root-causing effort. The goal of finding relevant process parameters to improve yield is similar to previous works [18]. However, the analysis is much more detailed than what has been previously proposed.

To the best of the author's knowledge, no prior work has attempted to study the underlying process of applying correlation analysis to yield optimization or to automate the data analytics steps of this process. From that aspect, the work in Chapter 3 is the first of its kind.

Because outlier screening is a popular approach for capturing parametric defects, many outlier methods have been proposed to achieve this purpose. Among them, some are already being commonly used by semiconductor manufacturers. For example, Part Average Testing (PAT) is a common practice applied to automotive product

lines [37]. PAT can include Static PAT (SPAT), Dynamic PAT (DPAT), Automotive Electronics Council DPAT (AEC), and Robust DPAT (RDPAT) [38]. A PAT method determines outliers based on the distribution of the measured values on a set of parts. Such methods will be referred to as *distribution-based* methods.

Another popular class of outlier methods includes methods such as Good Die in a Bad Cluster (GDBC), GDBC with Specific Bins (GDBC SB), and Bad Bin in a Bad Cluster (BBBC). These methods determine outliers by using the location information of failing dies on a wafer [38]. They can be called location-based methods.

There are also methods that utilize both location information and measured values to determine outliers. Popular examples include Nearest Neighbor Residual (NNR) [27][39] and Location Averaging (LA) [27][40][41].

Traditional multivariate outlier analysis include the use of Mahalanobis distance and the use of a linear regression model [42]. In recent years, multivariate outlier analysis for screening parametric defects has gained more popularity. For example, recent studies include methods using Principal Component Analysis (PCA) [10][12] and Support Vector Machine (SVM) one-class algorithm [43][8].

The methods used in the studies in Chapters 4 and 5 comprise univariate and multivariate methods, as well as methods that utilize location information. The specific details of the methods used are described in the respective chapter introductions.

1.4 The Issue of Subjectivity

Analytics can be viewed as an iterative search process that comprises the three steps pictured in Figure 1.3: (1) dataset preparation, (2) running an analytics tool, and (3) meaningfulness determination. In this process, dataset preparation and meaningfulness determination are largely empirical. From the raw data, an analyst decides how

to prepare the dataset for the tool to analyze. After the tool outputs a result, the analyst examines the result to determine if it is meaningful.

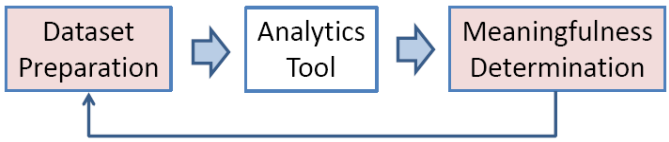


FIGURE 1.3: High Level Overview of Analytics Process

For *learning* to happen in analytics, two components are required [1]: data and domain knowledge. The need for domain knowledge to enable learning is an important detail which inspired much of the work in this thesis. The domain knowledge in the analytics process is responsible for most of the subjectivity. In the above steps, both preparing the dataset and determining whether results are meaningful require domain knowledge.

1.4.1 Subjectivity in Correlation Analysis

In correlation analysis, prior to feeding datasets to a correlation tool, an analyst must prepare the datasets. This dataset preparation step is to a large extent subjective to the analyst's experience and domain knowledge. For instance in yield optimization, when searching for correlation between a parametric test and a process parameter, just a couple of the decisions that must be made involve population selection and feature selection.

Population selection entails selecting a subset of data. Limiting the analysis to a subset of lots or a subset of wafers can often lead to entirely different results. Sources of systematic variation such as site-to-site variation or tester-to-tester variation can be dealt with in various ways. Separating the datasets is one possible solution, but it may lead to small datasets which could be statistically insignificant, especially when wafer sort is performed on 32 or more sites and the number of testers is large. Another

solution is to correct for variation by shifting the datasets, but this approach also has obvious drawbacks since the exact variation may be incalculable. Furthermore, if the period over which data was collected contained test revisions, process changes, or even gradual process shifts, clustering the data may be necessary to uncover meaningful results.

Feature selection may be even more convoluted. Data such as parametric tests and process parameters does not have an easy one-to-one mapping to run correlation on. Parametric tests are performed on every die on the wafer, while process parameters are measured on some site locations on the wafer. Typically the number of dies is much greater than the number of process parameter site locations. To make these datasets comparable, the analyst must decide how to create a one-to-one value mapping. One obvious approach is to perform correlation using the average values from each wafer, but this may not suffice to uncover an existing relationship between variables. Even when looking for correlation to a specific parametric test, using the actual test value, a normalized value, the number of failures, the number of parts with a particular value, or some statistic of the distribution all may be sound choices.

Many other decisions exist in preparing the datasets to feed into correlation tools and also in interpreting the results. Even if an analyst sets out to perform an exhaustive search based on all known perspectives, some perspectives may simply be unknown to the analyst. The effectiveness of correlation analysis is dependent on the analyst's domain knowledge. Chapter 2 demonstrates a real scenario where the perspectives applied by the analyst drastically affected the result of correlation analysis in yield optimization. Building on that result, the work in Chapter 3 proposes a methodology that can learn from an analyst's experience in order to reduce the subjectivity in subsequent analyses.

1.4.2 Subjectivity in Outlier Analysis

Recall that the general function of outlier analysis is to develop models that can identify outliers in data. However, the available outlier analysis methods require user input. In fact, each of the three components in Figure 1.2 requires the analyst to make subjective decisions. Thus, the performance of the resulting outlier models is dependent on choices made by the analyst.

Given a particular outlier method, a decision has to be made with respect to the samples in the base set. Common choices are using samples from one wafer or one lot as the base set. Although, in some cases it may make sense to use samples from a specific region of a wafer, samples tested on the same equipment, samples that meet some parametric condition, etc. An outlier in one base set may be an inlier in another base set. Hence, the choice of base directly impacts the effectiveness of the resulting outlier model.

Furthermore, a wide variety of methods are typically available at the analyst's disposal, complicating the issue with more subjective choices. The resulting outlier ranks produced by different methods may vary significantly. An analyst is often faced with the decision of choosing the best method for a particular problem. The effectiveness of methods can be assessed in many ways, for example by measuring the amount of variance reduction [39][40][41]. Also, it is common in practice for multiple methods to be employed. In that case, the data can be analyzed in order to devise the best sequence of outlier methods. For example, it has been shown that multivariate analysis methods can capture unique outliers that are not captured by univariate analysis methods [8][12].

The third component involves determining a threshold which separates parts into inliers and outliers in the outlier rank space. Threshold determination can be done in either a reactive fashion, where some set of failures is known, or in a proactive

fashion, where some yield reduction budget is established as an acceptable trade-off for improved quality.

In the reactive case, the known failures constitute any future failures that outlier analysis aims to capture at an earlier test stage, such as final test failures, burn-in failures, or customer returns. For final test failures, the tests where failures occur may not be covered in earlier test stages or failures may be caused by packaging. The existence of such false positives which are impossible to capture by outlier analysis leads to the success of the analysis being highly dependent on the analyst's ability to recognize such cases. As for burn-in failures and customer returns, the set of failures may not be sufficiently representative. The occurrence of these failures may be on the order of 1 PPM (parts per million) or even lower, which makes any resulting outlier models difficult to validate. Also, the complete set of failures can never be known because other similar parts could be missing from the set for a variety of reasons, such as potential customer returns no longer being used in the field or burn-in time being just short of exciting the failure.

In the proactive case, the optimal yield reduction budget may be difficult to establish. It is not uncommon for products to have at least three wafer sort stages and three final test stages, each with over 1000 tests. Therefore, even with a firm budget, the number of features in the test data is far too large to allow for an objective decision to be made about a global outlier ranking across features. Furthermore, as will be shown in Chapter 4, the performance of an outlier model can change as characteristics in the test data change over time. As such, outlier models with thresholds that meet a yield reduction constraint in some production data may result in far greater or lower yield loss in future production. Hence, model validation may be required to adapt to changes [44].

The subjectivities in outlier analysis can make it challenging to justify the outliers. In practice, justification can be supported by detailed analysis verifying that some of

the outliers are indeed bad parts. This task is both expensive and time consuming. There is also the concern that incomplete outlier sets lead the models to miss-classify bad parts as inliers. Lastly, recall from Figure 1.2 that the a threshold is decided based on an outlier ranking. This leads to an inherent limitation of outlier analysis which makes it unable to determine if there are no outliers at all. Additional insight must be provided by the analyst in order to decide that an outlier ranking contains no outliers. The subject of outlier model justification is explored further in Chapter 5.

1.5 Dissertation Organization

This dissertation contains self-supporting chapters, each with an overview, introduction, background, experimental setup, results, and a summary. It is suggested that Chapter 2 be read before Chapter 3 and that Chapter 4 be read before Chapter 5 if the reader wishes to follow the thought process behind the developments in the research. However, any chapter can be read independently based on reader interest. The rest of the dissertation is organized as follows. Chapter 2 presents an example where clear added benefit of data analytics is demonstrated for a yield optimization problem. Chapter 3 discusses an approach for learning how an analyst conducts the process of correlation analysis, using the result from Chapter 2 as the main supporting example. Chapter 4 introduces uncertainties that exist in outlier analysis and explores the generalization of outlier models with the goal of reducing those uncertainties. Chapter 5 proposes a concept called Consistency which allows for assessment of outlier models. Chapter 6 concludes the dissertation and discusses future research directions.

Chapter 2

Yield Optimization Using Advanced Statistical Correlation Methods

2.1 Overview

The work presented in this chapter introduces a novel yield optimization methodology based on establishing a strong correlation between a group of fails and an adjustable process parameter. The core of the methodology comprises three advanced statistical correlation methods. The first method performs multivariate correlation analysis to uncover linear correlation relationships between groups of fails and measurements of a process parameter. The second method partitions a dataset into multiple subsets to maximize the average of the correlations calculated based on the subsets. The third method performs statistical independence test to evaluate the risk of adjusting a process parameter. The methodology was applied to an automotive product line to improve yield. Silicon results are used to demonstrate how discovered process parameter changes led to significant improvement of the yield issue. This work is used to show the added value of data analytics in Test.

2.2 Introduction

Yield is one of the most important metrics to indicate the success of product manufacturing. Therefore, it is not unusual that efforts to improve yield continue into the mass production stage. In this work, yield optimization specifically refers to such efforts in production stage where mass amounts of test data become available and can be utilized to improve yield.

Due to process variations, yield is not a constant across wafers and lots. For example, Figure 2.1 illustrates a fluctuation of yield across wafers. The plot shows the probability density distribution of yield estimated based on 2000+ wafers. The chip is a sensor device for the automotive market, which contains a controller, sensors, and analog and RF components.

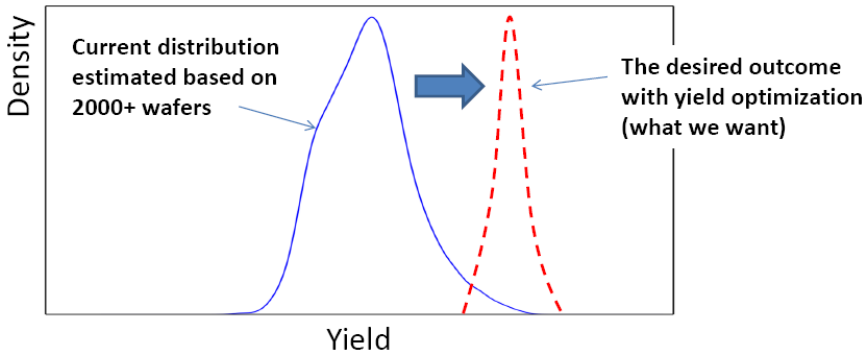


FIGURE 2.1: Illustration of yield fluctuation and our goal

Given Figure 2.1, it is desirable to push the yield distribution to the high end. In this sense, yield optimization can be thought of as both (1) pushing the mean of the yield to the right and (2) reducing the variance of the yield.

Because yield is such an important metric, multiple teams are in charge of improving it. For example, yield is a function of test. Hence, it is possible to improve yield by improving test (under the constraint that a measure of quality such as customer return rate is not worsened). From this end, note that result shown in Figure 2.1 was after

multiple test revisions that did not succeed in achieving the desired yield optimization.

Yield can also be design dependent. It is noted that the result seen in Figure 2.1 was also after one design revision. Therefore, additional yield improvement from Figure 2.1 would represent added value to both the design and test efforts.

The third way to improve yield is by adjusting the process. In order to identify which process parameter(s) to tune, evidence is required to show strong correlation relationships between process parameter(s) and certain types of fails of interest. This task is carried out by another team which can be called the yield analysis team.

To search for a high correlation between process parameter and type of fails pairs, an intuitive methodology can be based on a flow of the following five steps:

1. Identify a type of fails to investigate.
2. Calculate the numbers of fails across N wafers as $\vec{x} = \{x_1, \ldots, x_N\}$.
3. Calculate the measured value of a selected process parameter across the N wafers, one value per wafer as $\vec{y} = \{y_1, \ldots, y_N\}$.
4. Calculate the (Pearson) correlation coefficient as in the equation below, where $\bar{x}$ and $\bar{y}$ are the mean of $\vec{x}$ and the mean of $\vec{y}$, respectively.

$$Corr(\vec{x}, \vec{y}) = \frac{\sum_i^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i^N (x_i - \bar{x})^2 \sum_i^N (y_i - \bar{y})^2}} \tag{2.1}$$

5. Rank the parameters by the correlation coefficients and identify the top parameters.

Consider step (1) above. In test, failing parts are organized into different test bins. Usually, similar categories of fails are grouped into the same bin. Hence, it is natural to analyze each bin independently. For example, Figure 2.2 depicts the average number

of fails for a list of test bins (left plot), and for the top three most failing bins, bins 26, 25, and 28, their wafer-to-wafer fail fluctuations over time (right plot).

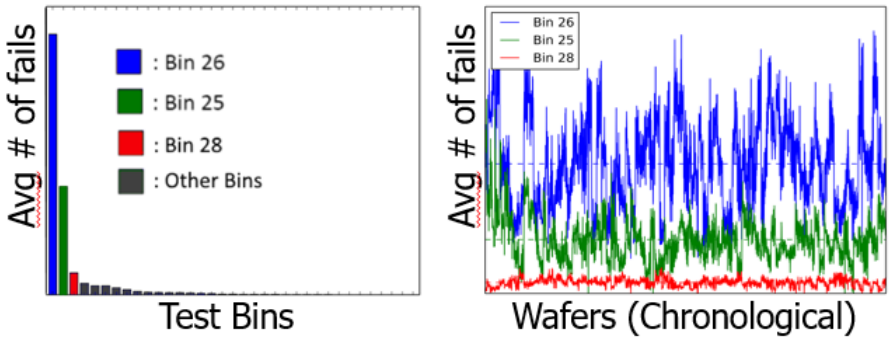


FIGURE 2.2: Bins of fails and their fluctuations

Given Figure 2.2, it is natural to consider bin 26 first, followed by bin 25. Suppose in step (1) bin 26 is chosen. Then, in step (2) the data vector $\vec{x}$ is extracted across the 2000+ wafers based on the failing dies (or "fails") recorded in bin 26.

Although partitioning the fails based on test bins makes sense, it is not the only way one can define a type of fails. For example, Figure 2.3 shows failing statistics based on individual tests in bins 26 and 25. As can be seen, tests A,B,C,D each have a significant number of fails. Hence, the type of fails can also be defined based on each individual test.

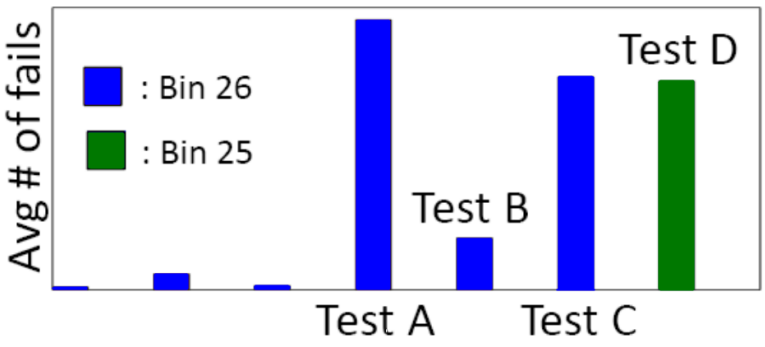


FIGURE 2.3: Failing statistics based on individual tests

Now consider step (3). Suppose t process parameters are measured for each wafer

(In this particular product, $t > 130$). Typically, these t measurements are repeated on multiple sites. Figure 2.4 gives an example of five sites and their locations on a wafer. The right plot shows the (wafer-to-wafer) fluctuation of the average measured value over the five sites for one process parameter. Given a process parameter, one can therefore extract the data vector $\vec{y}$ based on the average values.

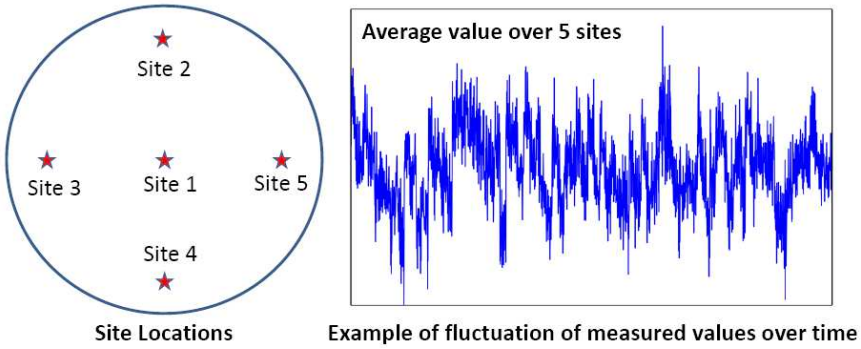


FIGURE 2.4: Measurement sites and a fluctuation example

Then, step (4) calculates the correlation coefficient $Corr(\vec{x}, \vec{y})$. Steps (3)-(4) can be applied to each process parameter to identify the one with the highest correlation. For example, the highest correlations found for bin 26, test A, test B and test C are depicted in Figure 2.5.

In these plots, the x-axis is the average value of the process parameter and y-axis is the number of fails. Each dot is a wafer. Similar results were found for other fail types (e.g. bin 25). Figure 2.5 basically shows that no strong correlation was found to support any potential process parameter adjustment. In other words, the intuitive correlation methodology described above had failed to uncover any useful information to support taking the path of process change to improve yield.

Figure 2.5 illustrates the starting point of this work. It is important to note that before this work, the yield analysis team had conducted extensive analysis of the test data and did not find a strong correlation. While their results were much richer than

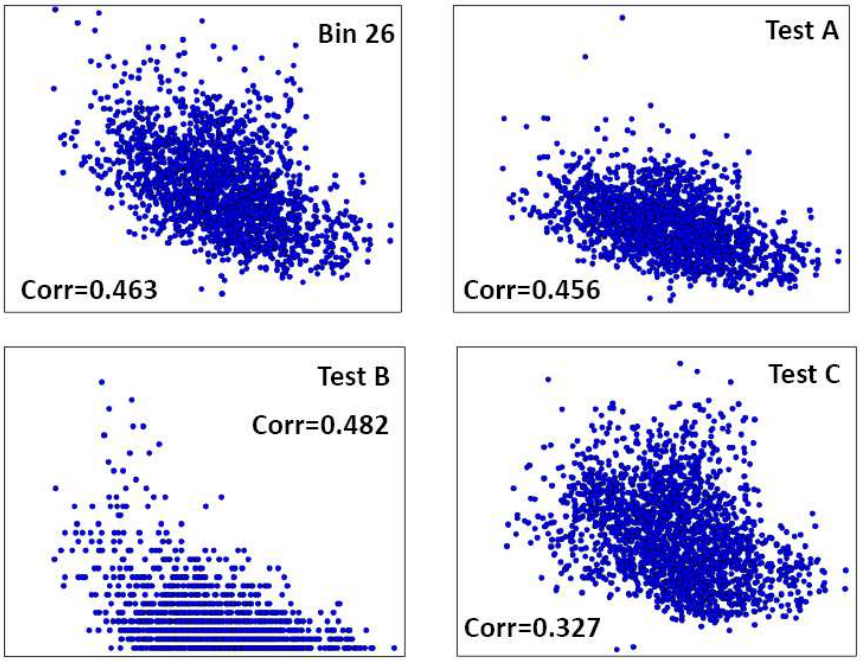


FIGURE 2.5: Illustrating the starting point of this work

those shown in Figure 2.5, the conclusion that no strong correlation could be found was the same.

The rest of this chapter is organized as below. Section 2.3 discusses potential issues with the intuitive methodology. Section 2.4 explains a multivariate correlation methodology and demonstrates its usefulness in uncovering a strong correlation which could not be found before. Section 2.5 describes a subset discovery problem formulated to enable finding additional strong correlations. Section 2.6 presents a risk evaluation method based on statistical independence test to assess the risk of a process parameter adjustment. Following the uncovered process parameter changes, section 2.7 summarizes the silicon results with significant yield improvement. Finally, section 2.8 provides a chapter summary.

2.3 Potential issues with the intuitive methodology

An obvious concern with the intuitive methodology is the straightforward application of Pearson correlation to evaluate statistical dependence. Pearson correlation coefficient, although popular, is not a robust statistic to test for dependence. For example, it is known that the correlation coefficient can be quite sensitive to strong outliers. Figure 2.6 illustrates such an example.

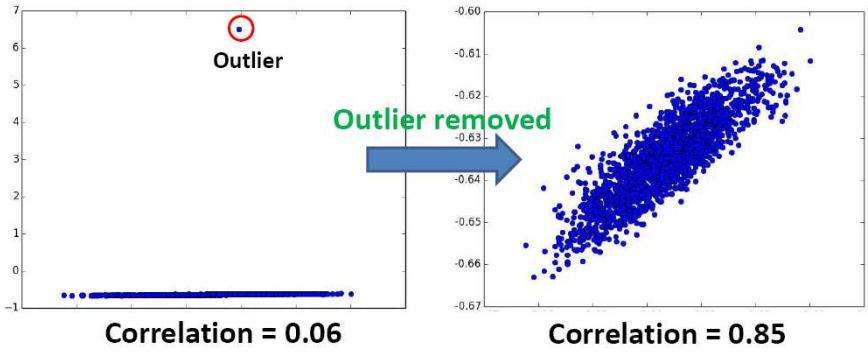


FIGURE 2.6: Correlation can be sensitive to outliers

The left plot shows that an outlier is far from the majority of the distribution. The correlation calculated based on this plot is 0.06. The right plot shows the result by removing the outlier (consequently, the scale of y-axis changes). The correlation becomes 0.85. The sensitivity to outliers can be one reason to use other statistics such as the rank correlation coefficients, e.g. Spearman's ρ or Kendall's τ. Alternatively, a preprocessing step can be taken to remove outliers.

2.3.1 Need for multivariate analysis

While there are many other alternatives for univariate statistical dependence test, a more fundamental issue is that the data used is inherently multivariate.

To apply a univariate analysis between $\vec{x}$ and $\vec{y}$, one has to define how to calculate $\vec{x}$ and $\vec{y}$ from the data. The correlation result can be dependent on the dataset

preparation. For example, each process parameter was measured on five sites. In the univariate analysis above, $\vec{x}$ was simply the average value of the five sites.

Figure 2.7 uses eight parameters P1-P8 to illustrate the variability across the five measurement sites. For each parameter, dots represent the correlation coefficients between measured values from all pairs of sites. Since there are five sites, there are 10 (pairwise) correlation coefficients shown for each parameter. In total, therefore there are 80 correlation coefficients (as shown in the x-axis) divided into eight blocks.

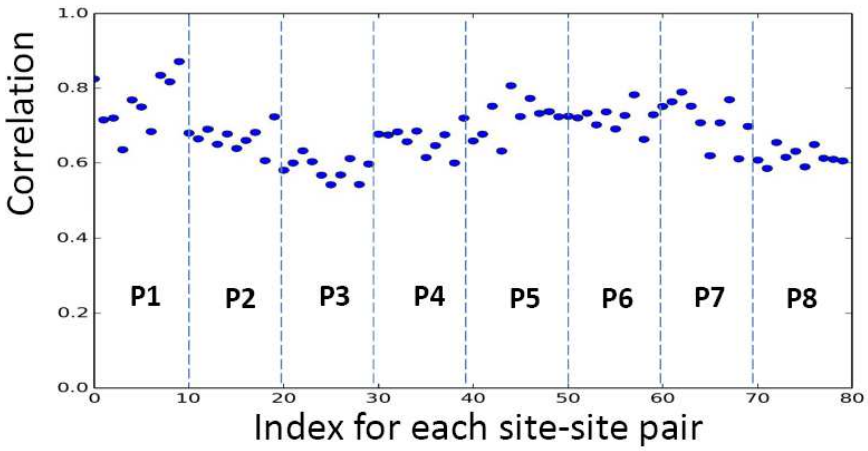


FIGURE 2.7: Examples of site-site pairwise correlations

Figure 2.7 shows that the correlations between two sites can range from below 0.6 to above 0.8. The main takeaway is that there can be significant variability across sites. When that is the case, treating the data from process parameters as a single average value may not be sufficient.

Furthermore, recall from Figure 2.3 that tests A-D are important due to their associated large numbers of fails. Figure 2.8 depicts the distribution of test values for test A and for test D.

Test A is a discrete test. Its values can fall into 6 categories. The left plot shows a histogram of the test values across one lot of wafers. Test D is a continuous tests. The

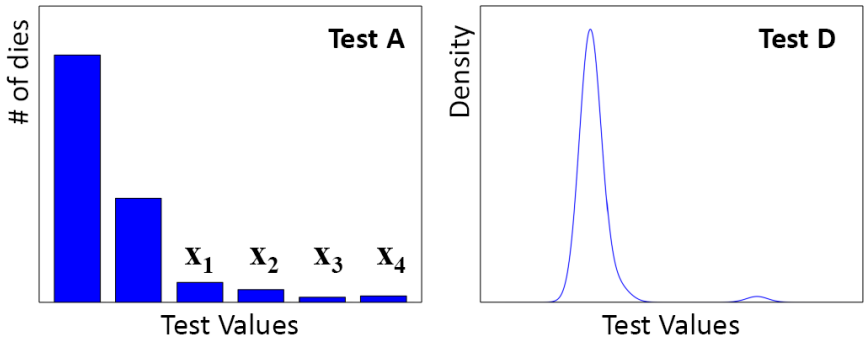


FIGURE 2.8: Discrete test A and continuous test D

right plot shows the probability distribution of the test values estimated based on one lot of wafers.

In the intuitive methodology, $\vec{x}$ is based on the number of fails. The number of fails does not include all the information contained in a test distribution as shown in Figure 2.3. For example, a process parameter may be correlated to only a subset of the fails (e.g. only having the X_4 value) or to the shape of a test distribution. For capturing these types of correlations, the correlation analysis needs to be extended to go beyond just using the number of fails as the target of the correlation.

For example, for test A the goal may be to correlate directly to a multivariate vector (X_1, X_2, X_3, X_4) as shown in Figure 2.3 (In general, the vector may be X_1-X_n for a large n). For test D, the goal may be to correlate to some characteristics of the distribution. Both demand a multivariate correlation analysis.

Figure 2.9 gives another reason to consider multivariate analysis. Figure 2.9 plots two wafer heat maps based on the number of fails in a single lot. Observe that test A fails concentrate on the edge while test D fails reside more on an inner ring. This raises the question: Would it be possible that a strong correlation exists only in a certain region of a wafer but not others?

One way to address this question can be to partition the wafer into multiple regions. A strong correlation may exist with each region individually or with a combination of

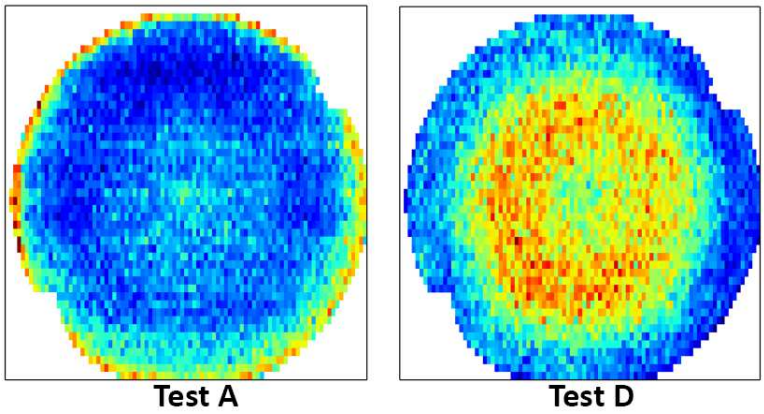


FIGURE 2.9: Examples of failing wafer heat map

multiple regions collectively. Again, this can be formulated as a multivariate correlation analysis problem.

2.4 Multivariate correlation and statistical dependence

It is well known that having no correlation does not imply statistical independence. In other words, finding no strong correlation does not mean that the type of fails and the process parameters have no strong relationship. The dependence relationship can still exist. A very low correlation coefficient only guarantees statistical independence when the joint probability distribution $P(x, y)$ is normal. In test data analysis, this is often not the case (see e.g., the left plot in Figure 2.8 is far from normal).

To go beyond correlation coefficient, one can follow the well established principles for measuring statistical dependence, proposed by Rényi who showed that one sound measure for statistical dependence is the following [45][46]:

$$\mathcal{Q}(P(x,y)) = \sup_{f,g} Corr(f(x), g(y)) \tag{2.2}$$

where f and g are Borel measurable and bounded functions. The notation "sup"

denotes the least upper bound. Hence, equation (2.2) basically denotes the maximum correlation across *all* possible functions f, g. Rényi showed that the quantity $\mathcal{Q}(P(x,y)) = 0$ implies statistical independence. The quantity $\mathcal{Q}(P(x,y)) = 1$ implies $x = h(y)$ or $y = h(x)$ for some function h, i.e. there is a strict dependence between x and y.

In equation (2.2), x and y are two random variables. Replacing them with two random vectors X and Y and also replacing the "sup" with the maximum "max", the following measure of dependence is obtained for two random vectors:

$$CC(X,Y) = \max_{f,g} Corr(f(X), g(Y)) \tag{2.3}$$

where f and g are functions that take a vector as input and output a real value. For example, when f and g are dot-product functions with weight vectors W_x and W_y, we have

$$CC(X,Y) = \max_{W_x, W_y} Corr(\langle W_x, X\rangle, \langle W_y, Y\rangle) \tag{2.4}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot-product of the two vectors.

In this case, the correlation calculated in equation (2.4) is the maximum correlation across all possible linear transforms denoted by W_x, W_y. Equation (2.4) is the traditional Canonical Correlation Analysis (CCA) [47]. $\langle W_x, X\rangle$ and $\langle W_y, Y\rangle$ represent the linear transforms based on the weights W_x and W_y. Hence, equation (2.4) basically looks for the maximum multivariate linear correlation.

A low $CC(X,Y)$ value in equation (2.4) does not guarantee statistical independence because W_x and W_y are linear transforms and the functions f and g in the original equation (2.3) can be non-linear. To extend CCA to consider non-linear transforms, kernel CCA (KCCA) applies the so-called "kernel trick" [48].

To apply the kernel trick, one starts with choosing a kernel function $k(X, Z)$ that measures the similarity between two vectors X and Z. A kernel $k()$ corresponds to a mapping function $\Phi()$ such that $k(X, Z) = \langle \Phi(X), \Phi(Z) \rangle$. The idea of KCCA is to apply CCA on the transformed vectors [49]:

$$KCC(X, Y) = CC(\Phi(X), \Phi(Y)) \tag{2.5}$$

The "kernel trick" corresponds to calculating equation (2.5) without explicitly using the mapping function $\Phi()$. Instead, only the kernel function $k()$ is involved in the computation. To explore non-linear correlations, a kernel is chosen for which the $\Phi()$ is non-linear. Equation (2.5) is called the kernel canonical correlation or the $\mathcal{F} - correlation$ [49].

This section discusses discuss how CCA can be used to find strong correlations beyond traditional correlation coefficient. Section 2.6 will discuss how KCCA can be used as a statistical independence measure for evaluating the risk of adjusting a process parameter.

2.4.1 Canonical Correlation Analysis (CCA)

Let $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_m)$ be two random vectors where each x_i and each y_j are random variables. In practice, the distribution of each x_i is measured by N sample points $\vec{x}_i = (x_{1i}, \ldots, x_{Ni})$. Similarly, each y_j is measured by N sample points $\vec{y}_j = (y_{1j}, \ldots, y_{Nj})$. Hence, this results in a data matrix S_x for X and a data matrix S_y for Y. This is illustrated in Figure 2.10 below.

Let $w_x = (w_1^x, \ldots, w_n^x)$ be a weight vector for X. In matrix multiplication "$S_x \times w_x$," the weight vector transforms each sample vector $\vec{u}_i$ into a canonical value $c(X)_i$ as below:

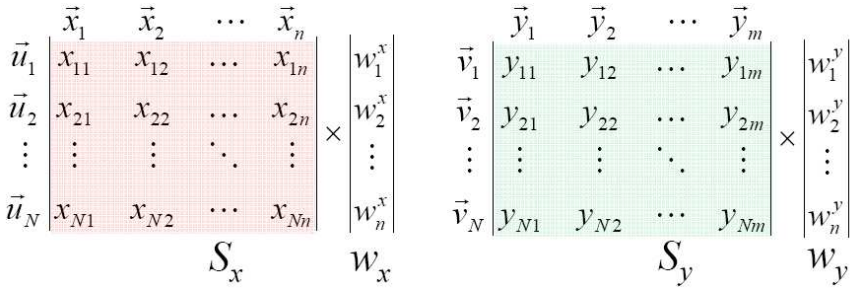


FIGURE 2.10: Data Matrices

$$c(X)_i = \langle w_x, \vec{u}_i \rangle = \sum_{k=1}^{n} (x_{ik} w_k^x) \tag{2.6}$$

Therefore, the result of $S_x \times w_x$ is a vector $(c(X)_1, \ldots, c(X)_N)$. Similarly, $S_y \times w_y$ is also a vector of N values. The correlation coefficient between the two vectors can then be calculated, denoted as $Corr(S_x w_x, S_y w_y)$. Then, the sample canonical correlation between X and Y is defined as

$$CC(X,Y) = \max_{w_x, w_y} Corr(S_x w_x, S_y w_y) \tag{2.7}$$

$$= \max_{w_x, w_y} \frac{\langle S_x w_x, S_y w_y \rangle}{\|S_x w_x\| \|S_y w_y\|} \tag{2.8}$$

S_x is an $N \times n$ matrix. w_x is an $N \times 1$ matrix (a column vector of size n). Hence, $S_x w_x$ is an $N \times 1$ matrix (a column vector of size N). On the other hand, $w_x' S_x'$, where $'$ denotes the matrix transpose operator, corresponds to the same vector as a $1 \times N$ matrix. Hence, using the notation $w_x' S_x'$, the numerator in equation (2.8) can be rewritten as $w_x' S_x' S_y w_y$ without using the dot-product $\langle \cdot, \cdot \rangle$ operator. Similar changes can be applied to the denominator to rewrite equation (2.8) as:

$$CC(X,Y) = \max_{w_x,w_y} \frac{w_x' S_x' S_y w_y}{\sqrt{w_x' S_x' S_x w_x}\sqrt{w_y' S_y' S_y w_y}} \tag{2.9}$$

$$= \max_{w_x,w_y} \frac{w_x' C_{xy} w_y}{\sqrt{w_x' C_{xx} w_x w_y' C_{yy} w_y}} \tag{2.10}$$

where $S_x' S_y = C_{xy}$ denotes the sample covariance matrix between X and Y, $S_x' S_x = C_{xx}$ denotes the sample covariance matrix for X and $S_y' S_y = C_{yy}$ denotes the sample covariance matrix for Y.

Because scaling on the weight vectors does not change the result of equation (2.10), the problem can be solved by maximizing the nominator $w_x' C_{xy} w_y$ subject to the constraints that $w_x' C_{xx} w_x = 1$ and $w_y' C_{yy} w_y = 1$. This is typically solved by applying the Lagrangian method. This leads to solving a generalized eigenproblem of the form $Aw_x = \lambda B w_x$ where $A = C_{xy} C_{yy}^{-1} C_{yx}$ and $B = C_{xx}$ [47].

Solving the generalized eigenproblem leads to a sequence of weight vectors for w_x. Then, each can be used to find the corresponding $w_y = \frac{C_{yy}^{-1} C_{yx} w_x}{\lambda}$. The number of weight vector pairs is equal to $\min(n, m)$, the smallest dimension between X and Y. The first weight vector pair gives the largest correlation. This can be called as the *1st CC component*. The second weight vector pair gives the second largest correlation (*2nd CC component*), and so on.

2.4.2 Analysis of test A in bin 26

Refer back to the test A plot in Figure 2.8. Let $X = (X_1, X_2, X_3, X_4)$ be the random vector as shown in the plot. Each X_i is a random variable, representing on each wafer the number of dies whose test A values fall into the X_i category. For test A, test values of X_1-X_4 are considered as failing. The other two are passing.

Given a process parameter P, let $Y_P = (S_1, S_2, S_3, S_4, S_5)$ be the random vector denoting the measured values on the five sites (see Figure 2.4). Then, CCA can be run on (X, Y_P). This can be carried out for each process parameter P to determine which one has the highest canonical correlation to X.

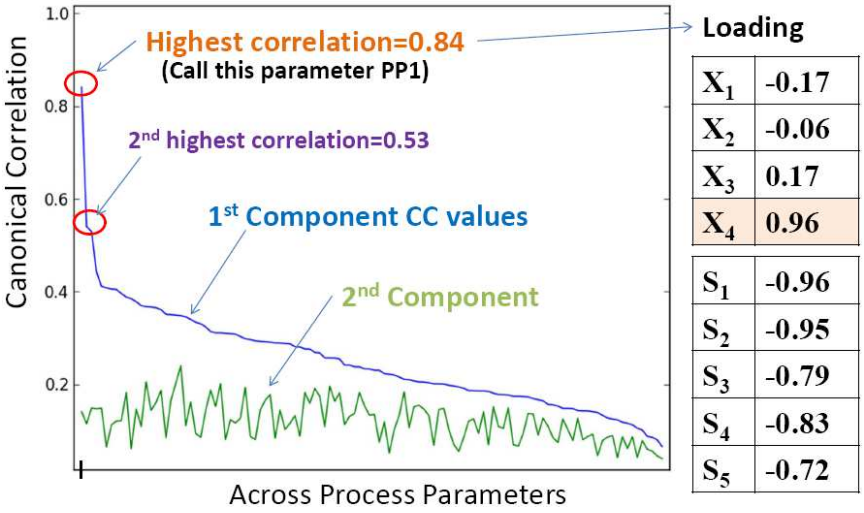


FIGURE 2.11: CCA results for Test A

Figure 2.11 plots the canonical correlations based on the first two CC components for all process parameters under consideration. Observe the correlations of the 1st CC components are consistently (much) higher than the correlations of the 2nd CC components. Also, the highest canonical correlation is 0.84, indicating a strong correlation between test A fails and the first process parameter shown in the plot (call this process parameter PP1). The second highest correlation to X is 0.53 which is based on the second process parameter in the plot.

The table on the right of Figure 2.11 shows the *loadings* for each random variable. To understand what a loading is, suppose $\vec{x}_1$ is the column vector denoting the N sample values measured on the random variable X_1. The *loading* in CCA for X_1 is simply the regular correlation coefficient between $\vec{x}_1$ and the transformed vector $S_x w_x$, i.e. loading$(\vec{x}_1) = Corr(\vec{x}_1, S_x w_x)$.

In Figure 2.11, it can be seen that the loading for X_4 is 0.96 that is much higher than the loadings for other X's. This indicates that the canonical correlation 0.84 is contributed more from the X_4 variable than from other X variables. In other words, it is likely that the X_4 type of fails by themselves have a high correlation to the PP1 parameter. On the Y variables, all sites have a fairly large loading. Hence, measurements from every site contribute to the 0.84 correlation result.

Let $S_x w_x$ and $S_y w_y$ be the transformed vectors for X and Y based on weight vectors w_x and w_y, respectively. The left side of Figure 2.12 plots the values from $S_x w_x$ against the values from $S_y w_y$ to show how they correlate. The plot shows a clear linear trend.

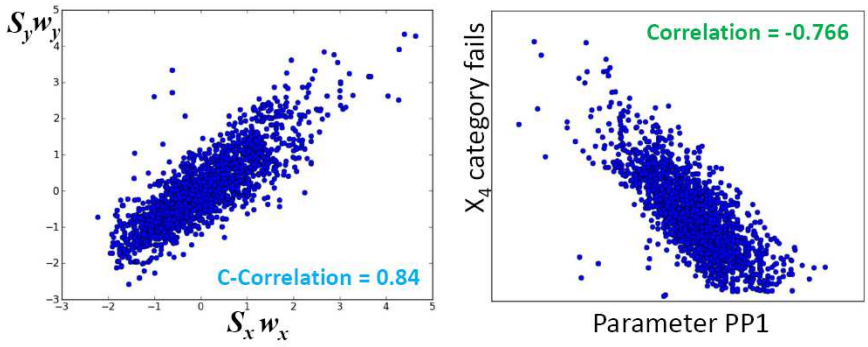


FIGURE 2.12: Further illustration of results shown in Figure 2.11

Then, the right plot of Figure 2.12 shows how the X_4 type of fails by itself correlates to the PP1 parameter. The x-axis is the average measured values for PP1 from the five sites (each dot is a wafer). As expected, a high correlation is observed with a negative correlation coefficient -0.766 between X_4 and PP1.

2.4.3 X_1, X_2, X_3 types of fails (removing X_4 fails)

Result in Figure 2.11 not only shows that X_4 type of fails are highly correlated to PP1, but also shows that no other high correlation can be found for X_1, X_2, X_3 fails individually or collectively. This is because (1) no high correlation (e.g. > 0.7) is found for other parameters (2) the 2nd component found for the PP1 parameter is quite low.

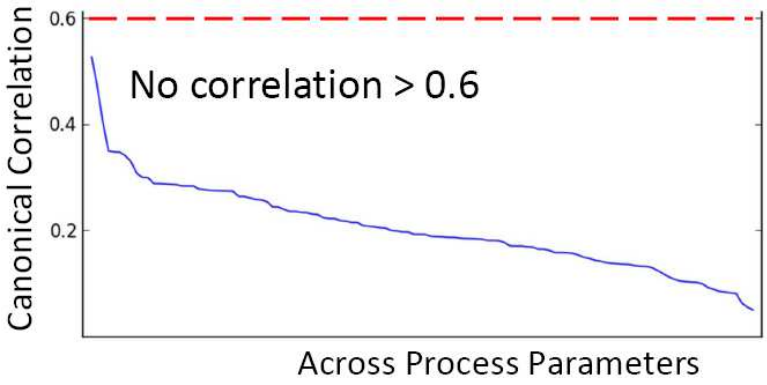


FIGURE 2.13: CCA with new random vector $X = (X_1, X_2, X_3)$

To verify the result, X_4 type fails were removed from the analysis to let $X = (X_1, X_2, X_3)$. CCA was then reran on the new X. The 1st CC component result is shown in Figure 2.13, showing that no high correlation is found across all process parameters.

2.4.4 Analysis of test D (Bin 25)

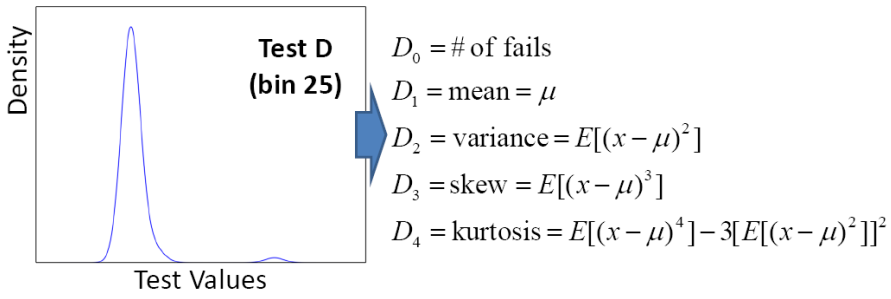


FIGURE 2.14: Encoding a distribution into a multivariate vector

Refer back to Figure 2.8 where the distribution of test D (bin 25) is shown. Every wafer can be represented by such a test distribution of test D estimated based on all dies on the wafer. Figure 2.14 shows a way to encode the characteristics of the distribution into a vector of five quantities, i.e. $X = (D_0, D_1, D_2, D_3, D_4)$. Then, CCA is ran with

on this X and each process parameter vector Y_P. Note that the encoding here was for the entire distribution, not just for the failing distribution.

The left plot of Figure 2.15 summarizes the CCA result where the highest canonical correlation found is 0.82. It is interesting to note that this highest correlation is based on the same parameter PP1 found in Figure 2.11. For comparison, the right plot of Figure 2.16 shows that univariate correlation between the number of test D fails (test D is the only test with bin 25) and the average PP1 value is only 0.305.

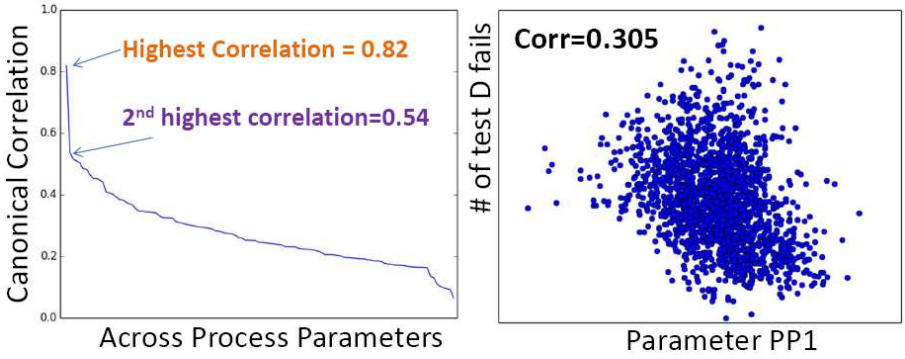


FIGURE 2.15: Canonical Correlation vs. Pearson Correlation

Figure 2.15 indicates that PP1 is highly correlated to some characteristics of the test D distribution, but not highly correlated to the number of test D fails.

After examining the CCA loadings for D_0 to D_4, they revealed that the two highest loadings were -0.74 for D_1 (the mean) and -0.91 for D_2 (the variance). This indicated that the PP1 was mostly negatively correlated to the mean and variance of the test D distribution. To confirm, Figure 2.16 shows the scatter plots for D_1-vs-PP1 and D_2-vs-PP1 with their respective Pearson correlation coefficients. The negative correlation trends can be observed in both plots.

2.4.5 Summary of the first finding - parameter PP1

In the above CCA analyses, CCA was used to identify a high correlation scenario. Then, the loadings were analyzed to identify the variable(s) contributing the most to

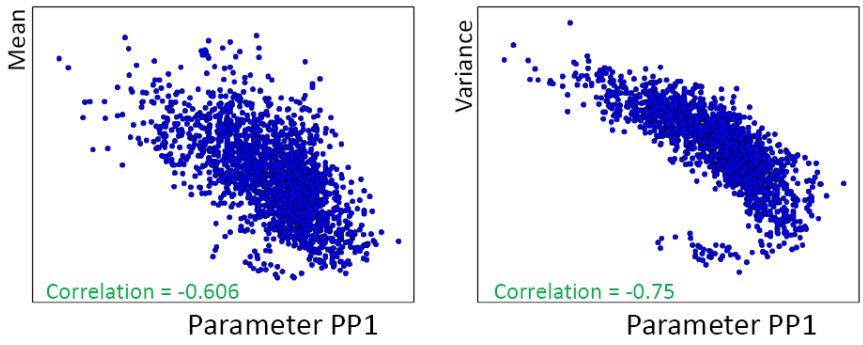


FIGURE 2.16: PP1 is correlated to the mean and variance of bin 25

the high correlation. Lastly, univariate correlation was used to confirm the findings.

Results from Figure 2.12 and Figure 2.16 both suggest to increase PP1 for yield improvement. With Figure 2.12, the expected impact would be to reduce the number of X_4 fails. With Figure 2.16, the expected impact would be to decrease the mean and variance of the test D distribution, resulting in fewer fails because the test limit is set on the right of the distribution.

2.4.6 Note on applying CCA in location-based analysis

Figure 2.9 earlier shows that fails of a certain type can distribute unevenly across the wafer. For example, in an analysis, one may desire to separate the fails close to the center of the wafer from the fails on the edge of the wafer.

To illustrate how CCA can be applied to analyze location-based correlations, Figure 2.17 gives an example of partitioning the X_4 type of fails in bin 26 into two groups, the inner group and the outer group. Let I be the number of X_4 fails in the inner group and O be the number of X_4 fails in the outer group. Let the random vector $X = (I, O)$ in CCA.

Running CCA on X and Y_{PP1} for process parameter PP1 gives canonical correlation 0.8, higher than the correlation -0.766 shown in Figure 2.12. Nevertheless, Figure 2.17 shows that the individual Pearson correlations are -0.754 and -0.745 for I-type of fails

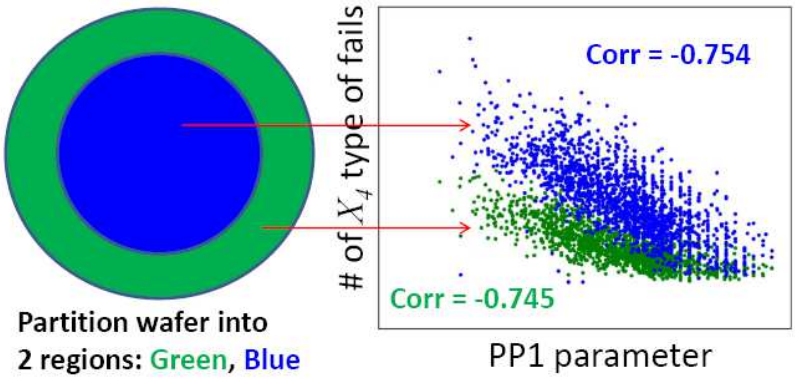


FIGURE 2.17: Partitioning X_4 type of fails based on their locations

and for O-type of fails, respectively. These results are comparable to -0.766 but not higher. Hence, this particular location-based CCA does not improve the correlation result.

2.5 The subset discovery problem

Section 2.4.3 shows that with the CCA method, no high correlation can be found to account for X_1-X_3 types of fails in bin 26. In this section, a subset discovery problem is formulated, and it is shown that solving this problem can enable the analysis to find additional strong correlations.

Given two random vectors X, Y, suppose X, Y are measured through a dataset S of N wafers. Let $S_1, S_2, \ldots, S_k$ denote a sequence of subsets of S where each $S_i \subset S$ and for any $i \neq j$, $S_i \cap S_j = \phi$. Let $|S_i|$ denote the size of the subset S_i. Let $CC(X, Y)_{S_i}$ denote the canonical correlation of the 1st CC component based on the wafers in subset S_i, i.e. the highest canonical correlation found. Then the subset discovery problem can be stated as the following:

Subset Discovery Problem:

$$SCC(X,Y)_{\lambda,\eta} = \max_{S_1,S_2,\ldots,S_k} [\frac{\sum_{i=1}^{k} CC(X,Y)_{S_i}}{k}] \tag{2.11}$$

subject to $|\bigcup_{i=1}^{k} S_i| \geq \lambda|S|$ and $\forall S_i, |S_i| \geq \eta|S|$,

where $0 < \lambda < 1$ and $0 < \eta < \lambda$ are given parameters.

Let $SCC(X,Y)_{\lambda,\eta}$ be called the *subset canonical correlation* based on the user parameters λ and η.

Suppose $\lambda = 0.5$ (50%) and $\eta = 0.1$ (10%). The constraint $|\bigcup_{i=1}^{k} S_i| \geq \lambda|S|$ basically says that the total number of samples used in the calculation of SCC has to be no less than 50% of the size of S. On the other hand, the constraint $\forall S_i, |S_i| \geq \eta|S|$ says that the size of each subset cannot be less than 10% of the total number of samples in S. It is important to note that all subsets are disjoint. Let the two constraints be called the λ-*constraint* and the η-*constraint*, respectively.

2.5.1 Assumption for subset discovery to be useful

The subset discovery problem tries to find k disjoint subsets under the two size constraints, to maximize the average canonical correlation across the k subsets. This is based on the assumption that for some subset S_i, $CC(X,Y)_{S_i}$ is much higher than $CC(X,Y)_S$. In other words, the correlation relationship can become a lot more apparent when focusing on a particular subset (and the correlation relationship becomes blurred if the entire dataset is used).

Using less data can be better because the original dataset can contain noise. For example, the measurements at a certain period may be more noisy than others. The parameter λ allows the user to drop a portion of the data to maximize the resulting correlation. Another reason can be because there is a drift of the correlation relationship over time. Given two subsets S_i and S_j produced at different times, a strong

correlation relationship can be identified based on each subset individually but not on both subsets collectively. This drift property will be discussed in detail shortly when the results are presented.

2.5.2 Heuristic to approach the problem

The objective function in the subset discovery problem involves calculation of canonical correlations $CC(X,Y)_{S_i}$. Further, in search for the best subsets $S_1, \ldots, S_k$, it may be necessary to consider all possible partitions (that satisfy the constraints) of the set S. Exhaustively searching for the optimal answer can be overly expensive.

A straightforward heuristic is to incrementally find the subsets following a greedy approach. In other words, the heuristic finds a sequence of subsets $S_1, \ldots, S_k$ such that $CC(X,Y)_{S_1} > CC(X,Y)_{S_2} > \ldots > CC(X,Y)_{S_k}$. The heuristic can be described as repeating a two-step process:

1. Given S, find the subset S_i that results in maximum $CC(X,Y)_{S_i}$ where $|S_i|$ satisfies the η-*constraint*.

2. If the λ-*constraint* is not yet satisfied, let $S = S - S_i$ and repeat step (1); Otherwise, stop.

Let S_i and S_{i+1} be two consecutive subsets found by the heuristic. Let s be a wafer such that $s \in S_{i+1}$. Note that it is possible to have the situation where $CC(X,Y)_{S_i} + CC(X,Y)_{S_{i+1}} < CC(X,Y)_{S_i \cup \{s\}} + CC(X,Y)_{S_{i+1} - \{s\}}$. In other words, if the sample s is moved from S_{i+1} back to S_i, the sum of the two correlations improves.

It is important to note that if the algorithm to solve the maximization problem in step (1) is ideal, then it should be true that $CC(X,Y)_{S_i} > CC(X,Y)_{S_i \cup \{s\}}$. However, this does not mean that s should not be included in S_i because s may decrease the correlation based on S_{i+1} more than it decreases the correlation based on S_i. Therefore, the straightforward heuristic is not optimal.

Because computing the objective function in the subset discovery problem itself can be expensive, it is preferable not to follow a process that goes beyond the linear complexity. This preference justifies the use of the greedy heuristic. Hence, the objective function in step (1) is modified by introducing a regularization term based on the size of the subset.

In step (1), instead of finding a subset to maximize $CC(X,Y)_{S_i}$, the objective becomes to maximize $CC(X,Y)_{S_i} + \gamma\frac{|S_i|}{|S|}$. In other words, if adding more samples to S_i does not decrease the correlation too much, then those samples should be added. Notice that $0 < \frac{|S_i|}{|S|} \leq 1$ and $0 \leq CC(X,Y)_{S_i} \leq 1$. Hence, value ranges of the two terms are comparable. This means that the choice of γ would not be too far from 1. The optimal choice of γ can be determined experimentally.

2.5.3 Analysis of X_1-X_3 types of fails from test A

Recall that Figure 2.13 summarizes the best result found for X_1-X_3 types of fails from test A in bin 26, where no canonical correlation was found to be greater than 0.6.

TABLE 2.1: Subset canonical correlations for four parameters PP2-PP5 found to have high correlations to the X_1-X_3 types of fails

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
PP2	0.84	0.78	0.643	0.69	0.641	0.63	-	-	-
PP3	0.88	0.85	0.69	0.82	0.68	0.61	0.53	0.52	-
PP4	0.86	0.87	0.85	0.82	0.83	0.82	0.81	0.79	0.75
PP5	0.86	0.81	0.82	0.77	0.68	0.59	0.61	-	-

Table 2.1 summarizes the results of applying subset discovery to analyze X_1-X_3 types of fails. For λ-*constraint*, we set $\lambda = 0.5$. For η-*constraint*, we set $\eta = 0.0625$ which means using a minimum of 125 wafers in each subset for a total of 2000+ wafers. In the table, the S_i represents the ith subset found by following the greedy heuristic discussed above. Each number shown is the canonical correlation of the 1st CC component based on the particular subset.

Notice that for the same parameter, it is not always true that the correlation found with S_i is greater than or equal to that with S_{i+1}. For example, for PP2 the correlation found with S_4 (0.69) is higher than that found with S_3 (0.643). This is due to the regularization discussed above, where in each step we try to maximize the term $CC(X,Y)_{S_i} + \gamma\frac{|S_i|}{|S|}$ instead of just maximizing the canonical correlation $CC(X,Y)_{S_i}$.

2.5.4 Result illustration

Table 2.1 shows that the four parameters PP2-PP5 can be highly correlated to the X_1-X_3 types of fails. To illustrate why subset discovery is needed for finding these correlations, Figure 2.18 and Figure 2.19 show, for each parameter, a scatter plot based on selected two subsets (green and blue). The correlations shown in these plots are Pearson correlation coefficients between the number of fails in the subset and the average parameter measured value across five sites.

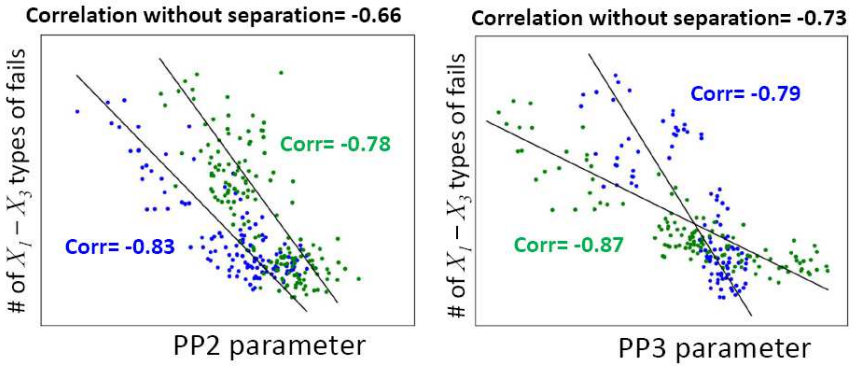


FIGURE 2.18: Subset discovery found two process parameters, PP2 and PP3, highly correlated to X_1-X_3 types of fails in bin 26

Consider the first plot in Figure 2.18, the two subsets individually correlate to the PP2 by -0.78 and -0.83. Collectively, the correlation drops to -0.66. The reason can be seen clearly from the plot that between the two subsets, there is a shift of the trend. Therefore, when all the data points are analyzed together, the trend becomes less apparent.

Examination of the two subsets showed that the wafers used in these two subsets were produced from two time periods. In other words, the shift of the correlation trend happened over time.

Similar shifts of trends can be observed for the PP3 plot in Figure 2.18 and for PP4 and PP5 in the two plots in Figure 2.19. In all cases, the correlations based on each subset of wafers are higher than the correlations based on the two subsets combined.

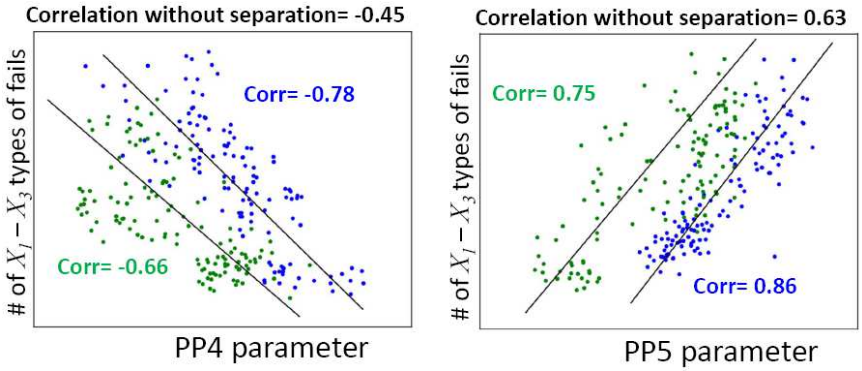


FIGURE 2.19: Subset discovery found two more process parameters, PP4 and PP5, correlated to X_1-X_3 types of fails in bin 26

Figure 2.18 and Figure 2.19 also show that subset discovery can be applied independently of CCA to find correlations. In the two figures, all correlations are based on Pearson correlation and it is sufficient to uncover high correlations once the appropriate subsets are identified.

2.5.5 Double check X_4 types of fails from test A

Earlier with Figure 2.12, the X_4 type of fails was established to be highly correlated to the parameter PP1. This is supported by the highest canonical correlation found, 0.84, as well as by the Pearson correlation itself, -0.766.

Table 2.2 shows the result of applying subset discovery to rerun the CCA analysis with $X = (X_1, X_2, X_3, X_4)$. The same subset discovery parameter settings were used, $\lambda = 0.5$ and $\eta = 0.0625$. Table 2.2 shows that all subset canonical correlations are

greater than the canonical correlation 0.84 found before. Four subsets give correlations above 0.9.

TABLE 2.2: Confirming strong correlation between X_4 fails and PP1

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
PP1	0.93	0.92	0.91	0.90	0.89	0.889	0.874	0.864

The left plot of Figure 2.20 shows results for X_4 type of fails from two subsets. Notice that individually the Pearson correlation coefficients are -0.91 and -0.85 which are much improved from the correlation coefficient -0.766 found before.

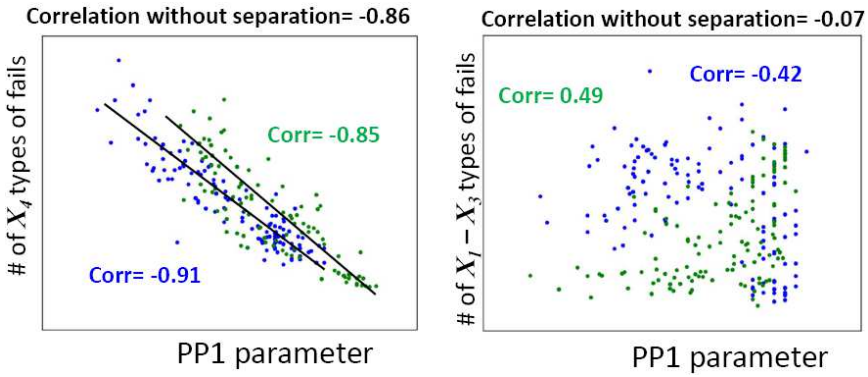


FIGURE 2.20: Subset discovery confirms X_4 type of fails highly correlated to PP1 while X_1-X_3 types of fails do not

For comparison, the right plot of Figure 2.20 shows results for X_1-X_3 types of fails. The result earlier shows that they do not have a high correlation to parameter PP1. The plot confirms the finding by showing two subsets with low (and opposite) correlations and with near zero combined correlation.

2.5.6 Summary of findings

Table 2.3 summarizes the correlation findings. Based on the results, the recommendation was: increasing PP1, PP2, PP3, PP4, and decreasing PP5.

TABLE 2.3: Summary of findings and supporting evidences

Para	Fail Type	Trend	Support
PP1	X_4 type, test A, Bin 26	Negatively correlated	Figures 2.12,2.20
PP1	Bin 25	Negatively correlated	Figure 2.16
PP2	X_1-X_3 types, test A, Bin 26	Negatively correlated	Figure 2.18
PP3	X_1-X_3 types, test A, Bin 26	Negatively correlated	Figure 2.18
PP4	X_1-X_3 types, test A, Bin 26	Negatively correlated	Figure 2.19
PP4	tests B,C, Bin 26	Negatively correlated	omitted
PP5	X_1-X_3 types, test A, Bin 26	Positively correlated	Figure 2.19

2.6 Risk evaluation

Silicon experiments are expensive. Therefore, before any recommendation of process change was implemented, the risk associated with that change had to be evaluated. For example, the result above shows that increasing PP1 would improve the yields in bin 25 and bin 26. However, it might also simultaneously increase the failing rates of other bins. One way to sure that this was unlikely to happen was by assessing the statistical dependence between PP1 and other bins. In other words, for risk evaluation, it was desirable to demonstrate that PP1 was statistically independent from all other bins.

The CCA and subset CCA methods described above could be used as a basis for risk evaluation. However, CCA and subset CCA being unable to find high correlation between PP1 and a test is not sufficient to conclude that PP1 and the test fails are statistically independent. As discussed in Section 2.4, this is because CCA only looks for linear correlations. Hence, to take the evaluation one step further, it is necessary to also consider *non-linear correlations*.

2.6.1 Kernel CCA (KCCA) looks for non-linear correlations

For non-linear CCA, the idea of *kernel CCA* was employed, as stated in equation (2.5) in Section 2.4 before. Figure 2.21 illustrates the basic principle of kernel CCA.

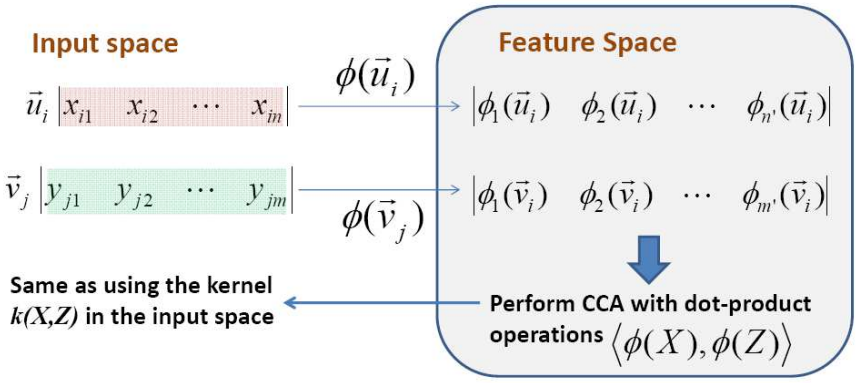


FIGURE 2.21: Illustration of kernel CCA

Let $k()$ be a kernel function and $\Phi()$ be corresponding mapping function where $k(X, Z) = \langle \Phi(X), \Phi(Z) \rangle$. Essentially, $\Phi()$ takes an input sample vector and maps it into another vector in the *feature space*. In Figure 2.21, for example, $\vec{u}_i$ is an n-dimensional sample vector (see also the matrix illustration in Section 2.4). $\Phi(\vec{u}_i)$ maps it to an n'-dimensional feature vector $|\Phi_1(\vec{u}_i), \ldots \Phi_{n'}(\vec{u}_i)|$ in the feature space.

Common kernels include the Gaussian kernel: $k(X, Z) = e^{-g\|X-Z\|^2}$ and polynomial kernel of degree d: $k(X, Z) = (\langle X, Z \rangle + R)^d$ for some constant R. For a Gaussian kernel, the dimensionality in the feature space is infinity ($n' = \infty$). For a polynomial kernel of degree d, the dimensionality $n' = \binom{n+d}{d}$ where n is the input dimension [48].

Kernel CCA is equivalent to performing the regular CCA in the feature space based on the mapped vectors. Refer back to equation (2.8) earlier for CCA formulation. The trick is to recognize that CCA is based on the dot-product operations between two vectors, i.e. such as $\langle X, Z \rangle$. Because $\langle \Phi(X), \Phi(Z) \rangle$ in the feature space is the same as $k(X, Z)$ in the input space, to perform CCA in the feature space, one can simply use the kernel operations $k(X, Z)$ in the input space to achieve the same purpose as illustrated in Figure 2.21. Hence, the mapping $\Phi()$ is never explicitly involved in kernel CCA. Rather, the computation is carried out using $k(X, Z)$ in the input space.

Let S_x be the data matrix for X containing N sample vectors $(\vec{u}_1, \ldots, \vec{u}_N)$. The

kernel matrix K_x is an $N \times N$ matrix $|k(\vec{u}_i, \vec{u}_j)|_{\forall i,j}$. Also let K_y denote the kernel matrix for Y. Notice that K_y is also an $N \times N$ matrix because there are N samples (wafers).

With the kernel trick, the kernel CCA can be stated as the following (α, β are N-dimensional vectors) [49]:

$$KCC(X,Y) = \max_{\alpha,\beta} \frac{\alpha' K_x K_y \beta}{\sqrt{\alpha' K_x^2 \alpha}\sqrt{\beta' K_y^2 \beta}} \tag{2.12}$$

Comparing equation (2.12) to the original CCA formulation equation (2.10), the only changes are that S_x is replaced with K_x and S_y is replaced with K_y [47]. Given a kernel with a non-linear mapping $\Phi()$, performing CCA in the feature space is therefore equivalent to maximizing the non-linear correlation in the input space.

2.6.2 Kernel CCA as a statistical independence test

It turns out that the formulation of equation (2.12) is not very useful in practice. This is because with a powerful enough kernel, it is almost guaranteed that $KCC = 1$. For example, with a well defined universal kernel [50] (e.g. a Gaussian kernel mentioned above is a universal kernel), it can be shown that the KCC result is always 1 [51], independent of the dataset. In other words, one can always find a mapping function $\Phi()$ complex enough to *overfit* the data so that the resulting correlation is 1.

The most popular way to resolve the overfitting issue is through *regularization* [51] - In equation (2.12) the objective function is changed by replacing $\sqrt{\alpha' K_x^2 \alpha}$ with $\sqrt{\alpha' K_x^2 \alpha + \gamma \alpha' K_x \alpha}$ and $\sqrt{\beta' K_y^2 \beta}$ with $\sqrt{\beta' K_y^2 \beta + \gamma \beta' K_y \beta}$. The user-input parameter γ controls the "complexity" of the linear transform functions used by CCA in the feature space. A small γ allows higher complexity and vice versa. It was proven that with regularization and universal kernels, $KCC(X,Y) = 0$ if and only if X and Y are independent [51].

2.6.3 Practical implementation of kernel CCA

Experimentation with the regularized kernel CCA found that in practice the results were hard to interpret. For example, depending on the choice of γ, kernel CCA may give a lower correlation than regular CCA which uses an unconstrained linear transform. This property is undesirable because for risk evaluation the kernel CCA is expected to always be more powerful than CCA, i.e. to always give an equal or higher correlation. A more practical implementation was therefore used, based on an idea proposing an alternative calculation [52].

The idea is to approximate kernel CCA by (1) running kernel Principal Component Analysis (KPCA) [53] to extract the first C principal components in the feature space and (2) running regular CCA based on the transformed dataset by the C principal components. In other words, in Figure 2.22 the kernel trick is applied to perform PCA in the feature space (kernel PCA), and CCA is then applied *directly* in the feature space by selecting only the first C kernel PCA components.

To illustrate the use of kernel CCA for dependence test, Figure 2.22 shows results based on parameter PP1. The CCA based analyses uncovered high correlations between PP1 and test A and test D, respectively. The analysis also showed that PP1 was not highly correlated to the most-frequent failing tests in bins 20, 28 and 30. Figure 2.22 shows how kernel CCA differentiates these two groups.

The x-axis represents the number C where the first C KPCA components are selected. Suppose X is an n dimensional vector $(X_1, \ldots, X_n)$. In the analysis, X is expanded to X' that is an $n + C$ dimensional vector $(X_1, \ldots, X_n, PC_1, \ldots, PC_C)$ where each PC_i is a KPCA component. Hence, for $C = 0$, it is the same as the regular CCA. As seen in Figure 2.22, as more KPCA components are used, the correlations become higher.

In Figure 2.22, the separation between the correlated cases and uncorrelated cases is

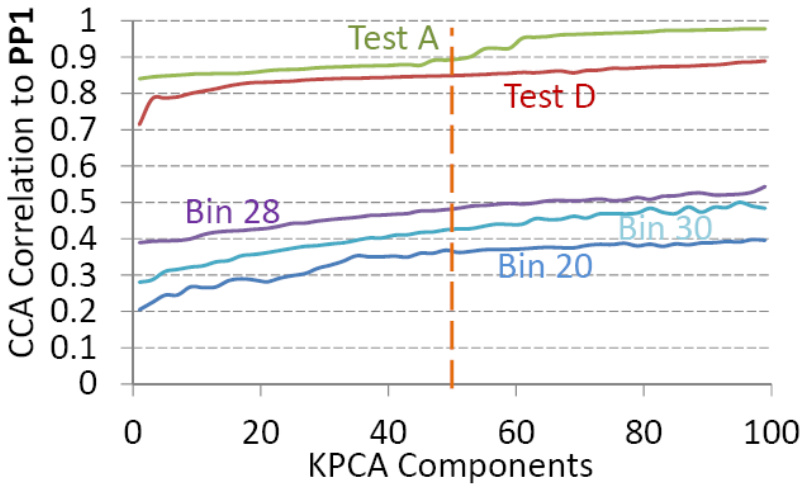


FIGURE 2.22: Kernel CCA risk evaluation on known results

clear across all selections of C. The number of KPCA components $C = 50$ was selected to apply the kernel CCA to check if there is a dependence between all other types of fails and PP1.

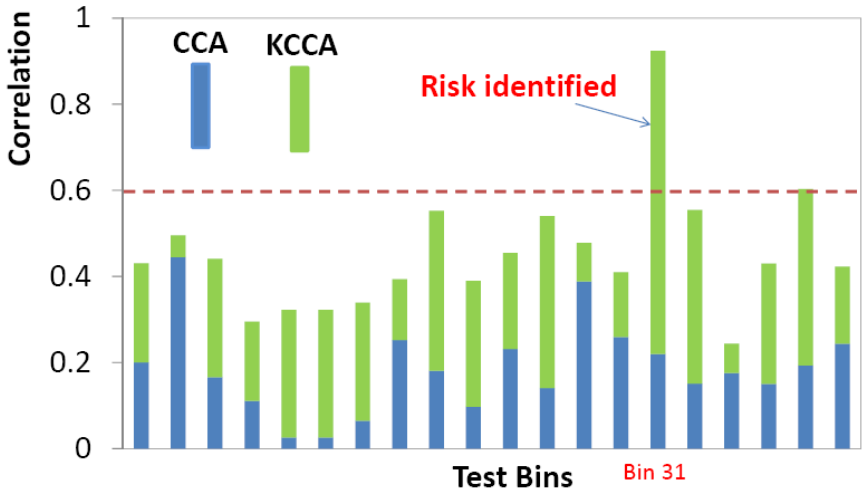


FIGURE 2.23: Risk evaluation with respect to adjusting parameter PP1

Figure 2.23 shows an example result of risk evaluation. This result illustrates the evaluation of the risk of adjusting PP1 by assessing the dependence between the result of a test and PP1. A test bin may comprise multiple tests. The figure shows the highest

KCCA correlation found in each test bin. In each case, it shows the correlation based on CCA and then the additional correlation based on the kernel CCA (with $C = 50$).

For all cases, the CCA correlations are low. For all but bin 31, the KCCA correlations are also not high. However, for bin 31, its CCA correlation is very low but KCCA correlation is very high - indicating a strong non-linear dependence between this test in bin 31 and the process parameter PP1.

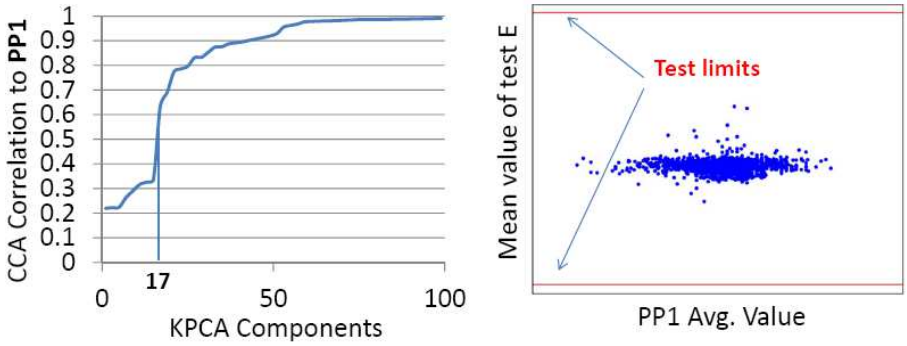


FIGURE 2.24: Detailed analysis of the test in bin 31 vs. PP1

Figure 2.24 provides more information on the test from bin 31. This test in bin 31 is henceforth referred to as test E. The left plot is similar to the plot shown in Figure 2.22. Observe that the KCCA correlation to PP1 increases significantly when the 17th KPCA component is included. The KCCA correlation increases to almost 1 as more components are added. This clearly indicates a strong non-linear correlation.

Since the dependence is non-linear, it is difficult to visualize. To contain the risk, the right plot shows a scatter plot where the y-axis is based on the average value of test E across each wafer. The plot shows that the distribution is not close to the test limits, i.e. the process capability index (C_{pk}) [54] is high. Hence, even though adjusting PP1 may somehow affect test E result, the risk of this adjustment causing a fallout on test E is not high.

The risk with test E was presented to the product team for further evaluation. It was determined that the association between PP1 and the devices tested by test E was

not high. In this case, the benefit of adjusting PP1 out-weighted the risk and hence, the adjustment was kept.

The same the kernel CCA based risk evaluation was used to assess the risk of adjusting other parameters PP2-PP5. A few other risky tests were found like that shown in Figure 2.23. However, all risky tests were contained either by showing a large margin of the distribution to the test limits (C_{pk}) and/or by domain knowledge from the product team. Although risk evaluation did not invalidate any of the recommended changes, it was an essential step to sign-off the silicon experiment.

2.7 Yield improvement based on silicon results

After the risk evaluation, findings from Table 2.3 were all accepted to design a split-lot experiment with multiple experimental wafers. The five parameter changes were implemented as three process changes, one for PP5 (call it ADJ #1), another for PP2-PP4 (call it ADJ #2), and the third for PP1. In the split lot experiment, change for PP1 was applied across the board. A first set of wafers was based on applying only ADJ #1 (and PP1 adjustment). A second set of wafers was based on applying only ADJ #2 (and PP1 adjustment). A third set of wafers was based on applying both ADJ #1 and ADJ #2 (and PP1 adjustment). Of course, lots manufactured previously without any of the changes were used for comparison.

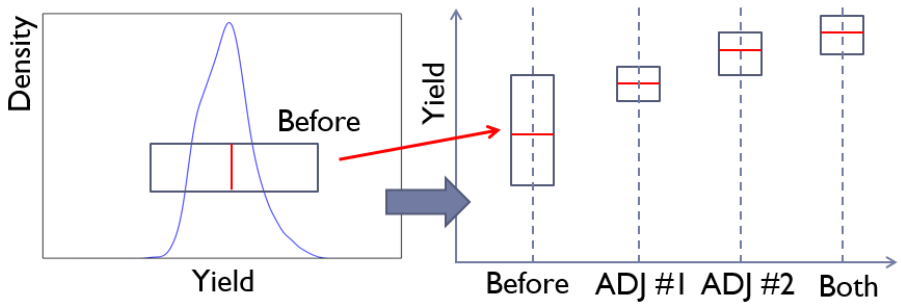


FIGURE 2.25: Silicon split-lot results show yield improvement

Figure 2.25 summarizes the result from the split-lot experiment. The right side contains a box plot showing the mean and range of the yield distributions before and after adjustments. It can be clearly seen that ADJ #1 and ADJ #2 each uniquely contribute to the yield improvement and together achieve the best yield result. After the split-lot experiment and confirmation of the yield improvement, the process changes were accepted and applied in production.

2.8 Summary

This work in this chapter presents a novel production yield optimization methodology based on three advanced statistical correlation methods: CCA, subset CCA and kernel CCA. The methodology was applied to optimize the production yield for an automotive product line. Silicon split-lot experiment confirmed the effectiveness of the findings by showing significant yield improvement and significant reduction of the yield fluctuation.

Recall that prior to employing the proposed methodology, unsuccessful yield optimization efforts were carried out by the test, design, and yield analysis teams. The silicon result demonstrates the clear benefit that applying data analytics had in this particular instance of the yield optimization application.

The success of the described analysis can be largely attributed to the recognition and treatment of correlation as a multivariate problem. In this sense, what contributed to the positive result most was simply employment of a perspective that had not been considered by in previous efforts. This finding inspired the work presented next in Chapter 3, which studies the development of new perspectives and proposes a methodology to learn from that process.

Chapter 3

Learning the Process for Correlation Analysis

3.1 Overview

An analytics process is subjective to the perspective of the analyst. This chapter presents a learning approach that models the process of how an analyst conducts analytics. The approach is applied in the context of correlation analysis for production yield optimization. The benefit is demonstrated by showing that learning from resolving a yield issue for one automotive product line can help resolve a yield issue for another automotive product line.

3.2 Introduction

Analytics has found many applications in test and has shown great promises. One example is yield optimization where, as shown in Chapter 2, analytics provides a clear added value to the efforts for improving production yield. As mentioned in Section 1.4, analytics can be viewed as an iterative search process that comprises three steps:

1. Dataset Preparation

2. Running an Analytics Tool

3. Meaningfulness Determination

These steps are visualized with some additional detail in Figure 3.1 .

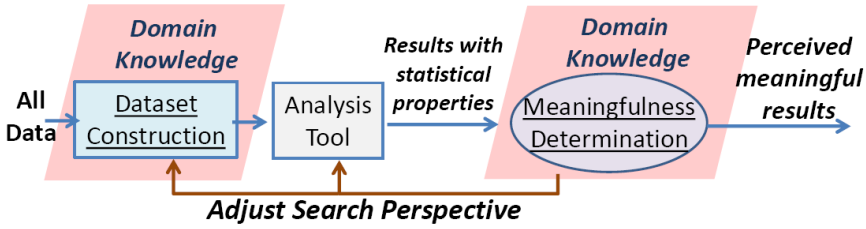


FIGURE 3.1: Analytics can be viewed as an iterative search process

An analyst has to decide how to construct the dataset to feed to the analysis tool. Then, the analyst has to interpret the result output by the tool to determine if it is meaningful. Therefore, the dataset preparation and meaningfulness determination steps are subjective to the experience and domain knowledge of the analyst.

For example, in the context of production yield optimization [55], an analyst desires to find a high correlation between a process parameter measurement (E-test) and a type of fails. Suppose for an E-test, its *average* measured values over multiple sites on a wafer, across n wafers, are $\vec{e} = \{e_1, \ldots, e_n\}$. On the other hand, the numbers of fails, based on a test bin, across the wafers are $\vec{f} = \{f_1, \ldots, f_n\}$. The analyst prepares a data file of two vectors $(\vec{e}, \vec{f})$. Then, with p E-tests and k test bins, the analyst prepares a dataset of $p \times k$ data files.

Each data file is fed into an analytics tool. For example, the analyst can use a statistical correlation tool from the Scikit-Learn Python library [56] to analyze each data file. This results in $p \times k$ correlation values. The analyst then examines these values to determine if any are meaningful.

If none of the correlation values are perceived to be meaningful, the analyst might decide to construct a different dataset. For example, instead of using each test bin as the basis for a data file, the basis can be each individual test.

In Chapter 2, ideas were presented for how to construct a dataset to search for high correlation between E-test and fails. Each construction represents one particular *perspective* for how the correlation might exist. In view of Figure 3.1, it is obvious that the effectiveness of the analytics depends on the set of perspectives the analyst has in mind. If the desired high correlation existed only in a perspective that the analyst had never thought of, then the analytics process would not find it.

Even though the work in Chapter 2 demonstrates that analytics helps improve the yield significantly, the above observation leads to two fundamental questions that arise when the analytics methodology is to be used for another product line:

- What if for another product line, high correlation only exists in a dataset that requires a perspective that was never considered when conducting the work in Chapter 2?
- What if the task of improving the yield for another product line is given to an engineer with little analytics experience who, for example, has no idea of the different perspectives examined in Chapter 2?

These two questions motivate the work presented in this chapter. Because of these two questions, it is desirable that different ways to prepare a dataset can be learned and generalized by a learning software. If this can be accomplished, the software can become a "surrogate" for the analyst in future tasks after learning how an analyst performs an analytics task. Therefore, this chapter presents a learning approach which aims to demonstrate the feasibility of such a learning software.

In particular, the learning approach will be shown to be able to learn from the analytics process for resolving the yield issue presented in Chapter 2 and apply the learning model to resolve another yield issue for another product line.

The rest of this chapter is organized as below. Section 3.3 provides a brief review of the work in Chapter 2. Section 3.4 discusses the learning problem studied in this work. Section 3.5 presents the approach for learning an analytics process. Section 3.6 explains the software design needed to bring process learning into the context of correlation analysis. Section 3.7 demonstrates the effectiveness of the approach. Section 3.8 discusses some limitations of the proposed approach. And finally, Section 3.9 provides a chapter summary.

3.3 Perspectives in yield optimization

A *perspective* is a particular way to construct a dataset. In test data analytics, *yield optimization* refers to the task of finding a high correlation between an a controllable parameter (e.g. E-test) and a set of failing dies. The analysis is wafer-based, meaning that two values are calculated for each wafer, one for E-test and the other for failing dies. Then, correlation is analyzed across a set of wafers. Using two values is for the case when one desires to use a standard correlation tool. If a canonical correlation tool [47] is used, two vectors of values are extracted for each wafer.

For simplicity, take standard correlation as the example. On the E-test side, one can have different ways to calculate the value representing the wafer. For example, a process parameter is measured on multiple sites, say 5. The value can be calculated based on taking the average of all 5 sites or a selected subset of sites.

On the side of failing dies, the choices can be many. A few examples that the value f_i used to represent a wafer W_i can take are:

- f_i is the number of fails due to a test bin (as mentioned before).

- f_i is the number of fails due to a test.
- f_i is the number of fails based on a particular test value.
- f_i is the number of fails based on a test value range.
- f_i is the mean value of a distribution of test values based on a test.
- f_i is the variance value of a distribution of test values based on a test.

In addition to the above diverse choices for the two values, a dataset can be constructed by taking two additional aspects into account: (1) a *spatial* aspect that restricts the population to a selected wafer region, (2) a *temporal* aspect that restricts the population to a set of selected wafers.

3.3.1 What contributed to the success in Chapter 2

The work in Chapter 2 presented a successful application of data analytics for resolving a yield issue for an automotive product line (a sensor product). Prior to the work, attempts for yield improvement were made through one design revision, multiple test revisions, and analytics to find high correlations, but all those attempts failed to improve the yield.

Note that the work in Chapter 2 presents other important methods (e.g. risk evaluation) than the diverse perspectives for dataset construction, which all contributed to the final success. However, a fundamental reason why the earlier analytics attempts failed but that work succeeded was indeed due to the fact that earlier attempts never analyzed the data from the perspectives the work employed.

For example, the highest correlation values found by the earlier analytics attempts were all below 0.5 which were not strong enough for the foundry to change their process. On the other hand, Figure 3.2 shows two example results (composed from results in Figures 2.12 and 2.16) and with (absolute) correlation values both above 0.75.

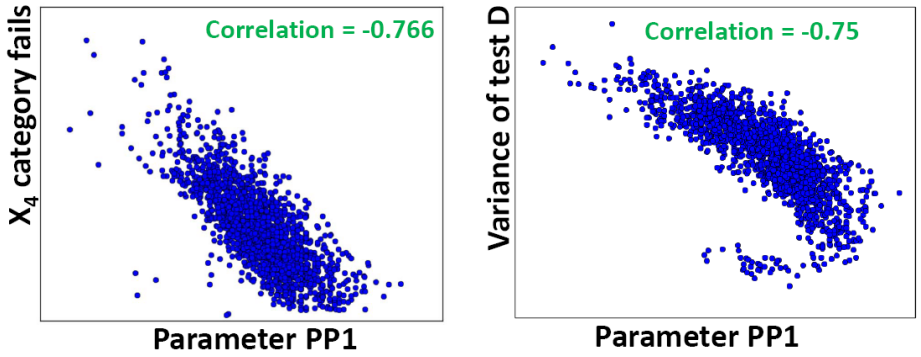


FIGURE 3.2: Examples of high correlations found

In the left plot (every dot is a wafer), the number of X_4 category of fails is correlated to the average E-test value from a process parameter PP1. Note that X_4 denotes a particular test value of a discrete test. In the right plot, the variance of measured values on another test D is also correlated to PP1. This example shows that the correlation can exist to some statistics of a test value distribution.

Figure 3.3 shows another example result by considering the *temporal* aspect (from result in Figure2.19). In this example, parameter PP5 is correlated to the number of X_1-X_3 categories of fails. The wafers are separated into two groups, colored as green and blue dots. The separation is based on the time of their production. Not all wafers are included. The separation improves the correlation values individually from their combined analysis result of 0.63.

As explained in Chapter 2, the high correlation results discovered were later translated into process adjustments which resulted in significant yield improvement on silicon. The adjustments were therefore adopted for mass production later.

3.4 The Learning Problem

As pointed out in the previous section, the main reason why the earlier analytics attempts did not discover the results as shown in Figure 3.2 and Figure 3.3 is that the

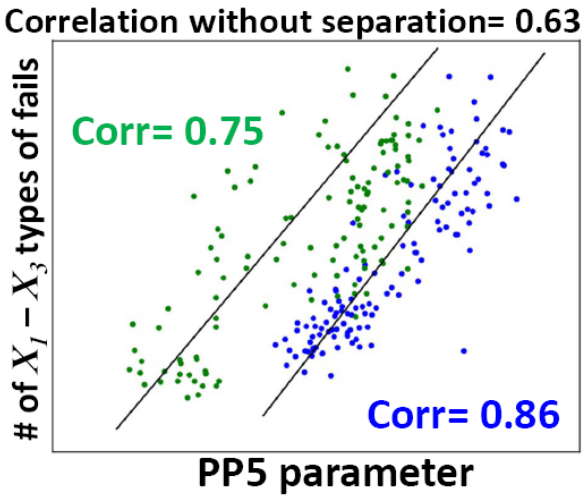


FIGURE 3.3: An example of uncovering the temporal effect

analysts conducting those attempts never constructed the particular datasets to look for those particular ways to correlate E-test and fails.

Suppose one desires to make a technology deployment of the work in Chapter 2 to all the product teams. The common practice today is to implement every perspective employed in the work into a software tool and deploy the tool. However, this approach is may not be sufficiently effective.

For a future task, it is possible that the required perspective to analyze the data is not in the set of the perspectives implemented. When that happens, the tool will fail. Then, because the product team usually does not understand the implementation of the software, they will ask the tool developer (the analytics expert) to debug and enhance the tool. This "centralized" approach puts all the burden on the expert.

Instead of providing a fixed set of perspectives, it would be more desirable for the tool to provide a set of toolboxes to enable the product team to conduct the search based on their own perspectives. More importantly, the tool could *record and generalize* from those perspectives and share that experience with other product teams. With this approach, the burden would be "distributed."

Developing such a tool demands answering the key question: "How can we learn (record and generalize) from someone's perspectives?"

3.4.1 Unsuccessful analytics trials

Learning the perspectives provides another advantage. It is common in practice that an analyst remembers the analytics process instances that lead to good results but forgets those that do not. With the proposed learning tool, all process instances (and their perspectives) can be recorded.

For example, for resolving the yield issue in Chapter 2, not only were different types of statistical correlation tried, but different attempts to establish an *association* relationship were also tried. Those association attempts were never reported in Chapter 2 nor the published in the related work [55] because they did not lead to successful results.

Unsuccessful example 1

For example, Figure 3.4 shows a heatmap association that led to an unsuccessful search. The left plot shows a heatmap constructed based on one lot of wafers. The color indicates the number of fails from the test bin that has the largest number of fails. Red means more fails.

The right plot shows the measured values of a frequency test. The interesting point to observe is that the wafer pattern exposed by the frequency test is similar to the failing heatmap. Figure 3.4 seemed to suggest that future search could be based on the frequency test. However, this was not successful.

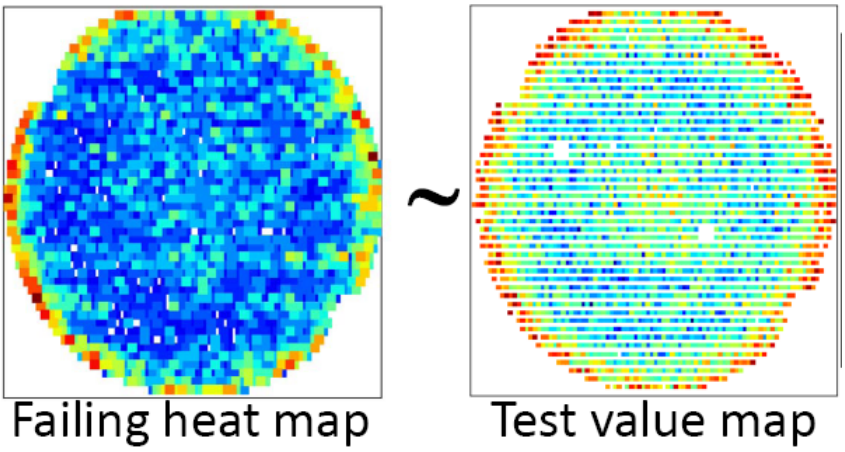


FIGURE 3.4: One example triggering unsuccessful search

Unsuccessful example 2

Figure 3.5 shows another unsuccessful example. The plot shows a 2-dimensional space with two E-tests, P_x and P_y. Each dot is a wafer, positioned by its average measured E-test values.

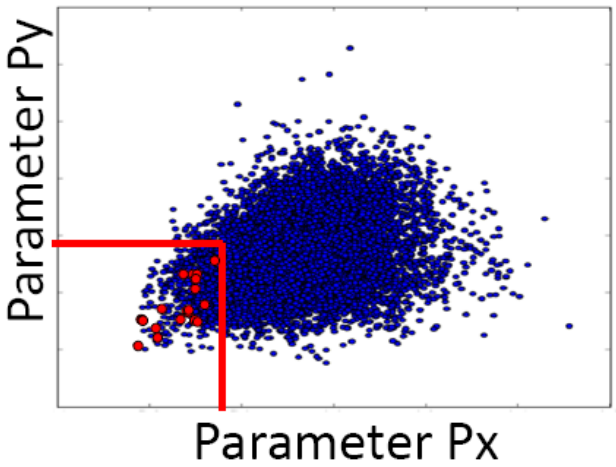


FIGURE 3.5: Another example triggering unsuccessful search

The red dots are the 20 wafers with the lowest yield. Notice that they concentrate on the bottom left corner of the plot. This plot associates lowest-yield wafers to the two process parameters. Hence, it was thought that future search could focus on these

two parameters. However, this again did not lead to successful resolution of the yield issue.

Even though the different types of association analysis were unsuccessful for that particular problem instance, it does not mean that they will not be useful for another task in the future. Hence, learning should also take such unsuccessful trials into account.

3.5 Learning the perspectives

As mentioned above, the objective of *learning* in this work is to *record* and to *generalize* the analytics process. The main subject of the learning is the set of *perspectives*, or *ways to construct a dataset*.

In order to learn perspectives, a way to represent a perspective is needed first. The learning algorithm and the effectiveness of learning depend on this representation.

The representation employed in this work treats each perspective as a sequence of *steps* that manipulate the data. If each perspective can be represented as a sequence of steps, or a *path*, then Process Mining (PM) [57][58] can be applied.

Process mining was originally motivated by the need to analyze logs from Workflow Management systems in business applications. It has its root in the early research of inductive inference [59] that studies the classes of *learnable* formal languages. In theory, the PM problem is similar to learning a finite state machine (regular language) from the machine's inputs and outputs.

Though extensive research has been done in the field of process mining and an open-source PM framework is available [60], the existing PM tools are not optimal for the yield optimization application. Accordingly, a tailor fit methodology is proposed, which serves a similar purpose to existing PM tools but is tuned toward learning the process of correlation analysis.

The input to this methodology is a *log* file which contains process instances, henceforth referred to as *traces*, corresponding to execution paths. When translated to a directed graph structure, these *traces* can be viewed as independent paths from START to END. Suppose there are five possible steps denoted by letters A through E, and there are two example paths, ACB and DCE, that have been executed. The *log* file is composed of the set {ACB, BCE}. Each instance in the set is a *trace*.

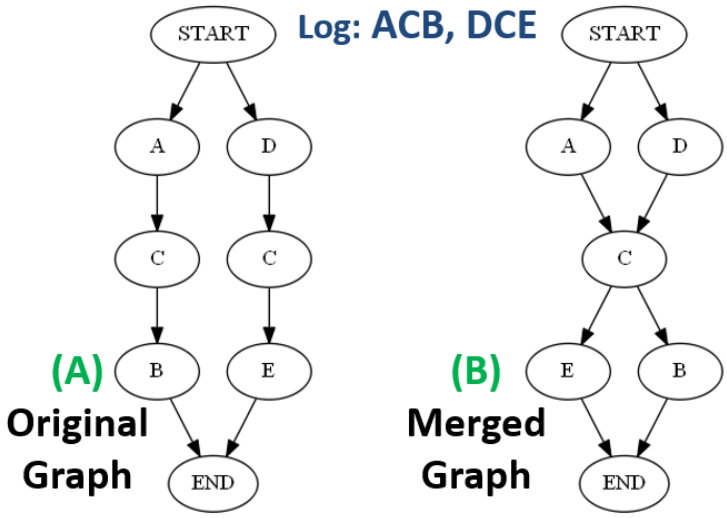


FIGURE 3.6: Simple state merging example

Figure 3.6 shows two possible process models that can be learned from the log. Model (A) basically records the two traces. Model (B), on the other hand, recognizes that step C is common in both traces. By merging step C into a single node, model (B) represents four traces, ACB, DCE, ACB, and DCE where the latter two are new. In this case, model (B) *generalizes* the log to include new traces.

When the log file becomes rather large, the number of possible merges grows quickly. This phenomena is illustrated using another simple example in Figure 3.7, where the input log contains 5 traces: {ABCF, AEF, ACE, BCDF, BDE}

In this example, model (A) again simply records all traces. Model (B) merges every step with the same name into a single node. Model (B) generalizes to have a total of 27

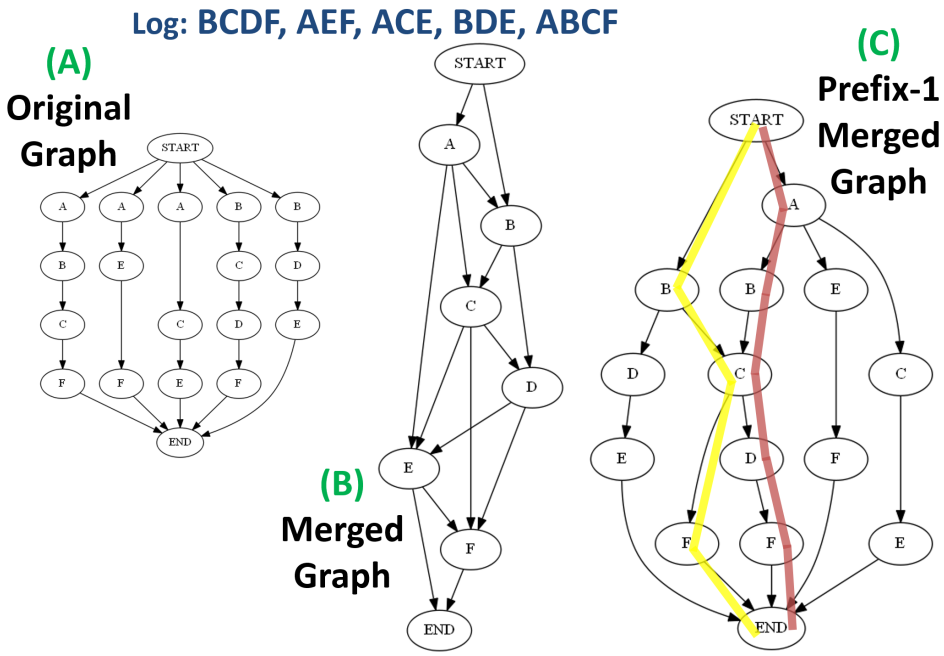


FIGURE 3.7: Another state merging example

traces in the model, including the original 5.

The rate at which the generalization scales brings up an important aspect of this process learning methodology. Recall that the goal of generalization is to aid an analyst by suggesting new analytics paths. However, an overwhelming number of such paths may not be desirable. A fundamental consideration in this approach is the trade-off between overfitting and underfitting. Because generalization is set as one the goals, maximum overfitting (i.e. only containing the input paths) is not useful. However, underfitting can get out of hand if it is not controlled.

One solution to underfitting lies in constraining when merging of process steps is allowed. With the simplistic state merging discussed thus far, any steps with multiple occurrences in the log were treated as having an identical state and were therefore merged into a single node. One way to constrain which steps are allowed to merge is to consider their preceding steps. This view redefines the underlying notion of identical

states from simply being instances of the same step to being instances of the same step with a matching *prefix*.

Consider Model (C) in Figure 3.7 which is based on a *prefix* rule. Suppose two partial traces are pX and qX where p and q are two sequences of steps each containing at least 1 step. Given a length l, let p_l be the last l steps in p and q_l be the last l steps in q. An l-prefix rule means that the two X nodes are merged only if $p_l = q_l$. Model (C) is obtained based on 1-prefix rule.

Notice in model (C) that every sequence is assumed to begin with a special step START and end at a special step END. Therefore, the three A steps in AEF, ACE, and ABCF, are merged into one node. However, this merging does not create any new traces. The two C steps in BCDF and ABCF are merged into one node because the steps before both C steps are B. This merging creates two new traces in the model, BCF and ABCDF, which are the only new traces in the model.

By using a prefix rule and by controlling the length l, one can control how many new traces are added into the model, i.e. how generalized the model is. This point will be shown later with an experimental result.

Implementation of a prefix rule poses a subtle issue – if two steps are merged, then the prefix for a step that follows the merged steps can be altered. For example, in Figure 3.7 model (B), after the C steps are merged, the 2-prefix of step E is altered. In the log, the 2-prefix of step E is only BC. After the merge, the 2-prefix of step E is {BC, AC}. Hence, a decision needs to be made for how a prefix rule applies if the prefix becomes a set.

In the implementation used for this work, if any pair of prefixes from two prefix sets are the same, the prefix sets are considered to be compatible and trigger merging of the two steps. Experimentally, this enables more (but not too many more) merges to take place, resulting in a more useful model. In practice, this makes sense because reaching a matching process node through the same sequence of steps confirms that

the nodes are likely to correspond to comparable states within the process.

The ordering of the merges is based on a partial ordering graph built based on the log. For two steps, X and Y, let X $>$ Y if Y only appears after X in the log. The merging begins with the more restricted graph like models (A) shown above. Then, it proceeds by following the partial ordering graph (i.e. following the topological ordering in the ordering graph).

Note that the usage of the prefix rule here is similar to that proposed in an existing algorithm [61]. However, this implementation is different because the existing algorithm [61] considers loops in the resulting model and this one does not. Also, this implementation additionally considers the prefix set merging rule.

Let X and Y be two steps and let p represent a sequence of steps. If a path XpY is in the log, it is possible that in order to execute Y, X has to be executed somewhere before the process. This type of *cross-steps dependency* can complicate process mining. This work does not consider step dependency other than that imposed by the prefix rule. The step dependency problem is avoided through careful design of the process steps.

3.6 Designing The Process Steps

The most challenging aspect of process mining is the design of the process steps. Each step essentially is a script (Python script in this case) that applies some manipulations of the current data and passes the resulting data to another step.

The importance of the step definition is similar to the importance of *feature selection* in machine learning. It has been widely shown that while the learning algorithm matters, the features to define the learning space can significantly impact the learning result. A notable recent example is the deep learning network [62] where much of the computation is for learning the importance of features.

Steps in process mining are like features. Hence, the design of the steps is a crucial part of the overall learning approach. An important stage for the development of the work is to recognize that, in correlation analysis, the end result is basically a figure as shown in Figure 3.8.

Figure 3.8 is important because it provides a view to design the steps. In this simple view, steps can be divided into 7 categories:

1. Defining the meaning of a dot
2. Defining the population of the dots
3. Defining the meaning along the x axis
4. Defining how the x value is calculated for a dot
5. Defining the meaning along the y axis
6. Defining how the y value is calculated for a dot
7. Optionally, defining how the dots are classified into colors

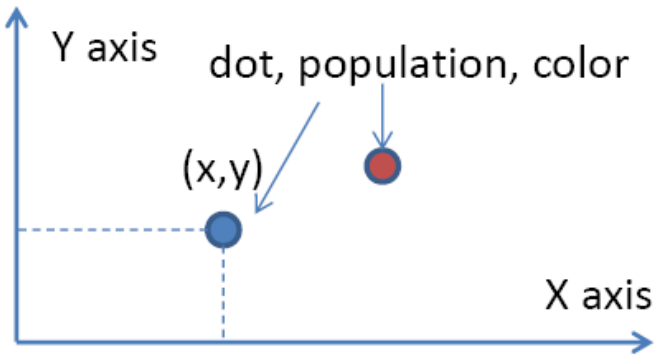


FIGURE 3.8: Dimensions to consider in designing process steps

These 7 categories of steps basically define how the dataset should be constructed to generate such a plot. Then, two additional categories of steps are defined:

1. Selection of the analysis tool to apply

2. Declaration of what type of result is to be reported (e.g. correlation value)

The following provides an example list of process steps to illustrate the software design:

- Dot type:
  PA (part)
  WF (wafer)
- Wafer grouping:
  WBTH (binary grouping using a threshold on the number of fails)
- Population (lot/wafer):
  LS (select all lots)
  L1S (select a lot)
  LSS (subset of lots)
  LSW (subset of wafers)
- Population (dynamic):
  CD (cluster lots by date)
  CLY (cluster lots by yield)
  CWY (cluster wafers by yield)
- Population (equipment):
  RS (restrict to a test site)
  RSS (restrict to a subset of test sites)
  RT (restrict to a tester)
  RST (restrict to a subset of testers)
- Population (location):
  RR (restrict to a ring on wafer)
  RC (restrict based on a radius from the center of wafer)
- X/Y axis (test):
  X-TS/Y-TS (select all tests)
  X-T1S/Y-T1S (select a test)

X-TSS/Y-TSS (select a subset of tests)

X-T1B/Y-T1B (select a test bin)

X-TSB/Y-TSB (select a subset of test bins)

- X/Y axis (E-test):

  X-P1S/Y-P1S (select an E-test from a site)

  X-PAS/Y-PAS (select an E-test from all sites)

  X-PSS/Y-PSS (select an E-test from a subset of sites)

- X/Y value (test):

  X-1V/Y-1V (use single test value)

  X-AV/Y-AV (use average value)

  X-WNF/Y-WNF (use number of failures)

  X-WST1/Y-WST1 (use statistics of the distribution)

  X-WNV1/Y-WNV1 (use number of parts with a particular test value)

  X-WNV/Y-WNV (use number of parts with value in a set of test values)

- XY coloring:

  X-TH/Y-TH (coloring based on dot property, e.g. pass/fail)

Three analysis tools are included: (1) statistical correlation (SC), (2) heatmap association (HA) (e.g. Figure 3.4), and (3) association analysis (AA) (e.g. Figure 3.5). Table 3.1 below shows how the above process steps can be concatenated into traces that produce the figures shown earlier in this chapter.

TABLE 3.1: Traces used for producing figures

Figure produced	Trace taken to produce the dataset
Figure 3.2 (left)	[WF, LS, X-PAS, X-AV, Y-T1S, Y-WNV1, SC]
Figure 3.2 (right)	[WF, LS, X-PAS, X-AV, Y-T1S, Y-WST1, SC]
Figure 3.3	[WF, CD, X-PAS, X-AV, Y-T1S, Y-WNV, SC]
Figure 3.4	[PA, L1S, X-TS, X-WNF, Y-T1S, Y-WNF, HSC]
Figure 3.5	[WF, WBTF, LS, X-P1S, X-1V, Y-P1S, Y-1V, AA]

Take Figure 3.2 (left) as an example. The first step "WF" defines each dot as a wafer. The second step "LS" defines the population to comprise all wafers. Then, "X-PAS" defines the x-axis to be based on values of E-test from all sites. The "X-AV" defines

the x value to be the average. Similarly, "Y-T1S" defines the y-axis to be a test. Then, "Y-WNV1" defines the y value to be the number of parts having a particular test value. Lastly, the resulting dataset is analyzed by a statistical correlation tool (SC).

In this design, steps such as "X-PAS," "Y-T1S," and "Y-WNV1" involve implicit enumeration across all possible choices. Hence, a trace involving them constructs a number of datasets through these implicit enumerations.

Note that adding new steps to the above set is straightforward. For example, if we desire to include Canonical Correlation (CC) [47] in the analysis, we just need to include new steps to define the x value and the y value. For instance, instead of a single x value, the x would be defined as a vector of values. Also, a new analytics step CCA would be added.

3.7 Applying PM Model

The product for the study of the second yield issue is an automotive part that operates in the 76-77 GHz band allocated for vehicular radars on an unlicensed basis (see e.g. FCC document [63]). To meet the specification, packaged chips are tested in cold temperature by operating at 76 GHz and in hot temperature by operating at 77 GHz. In both conditions, the voltage required to drive the oscillator is measured. An upper limit for the hot testing and a lower limit for the cold testing are set for the measured voltage. Unexpected yield drop is observed on some assembly lots with the cold and hot temperature tests.

Final test data are organized by assembly lots. Using each chip's ECID, the data are reorganized into their production lots. Figure 3.9 shows wafer-to-wafer variations of the test values at cold and hot temperatures. There are 175 wafers arranged by their production lots.

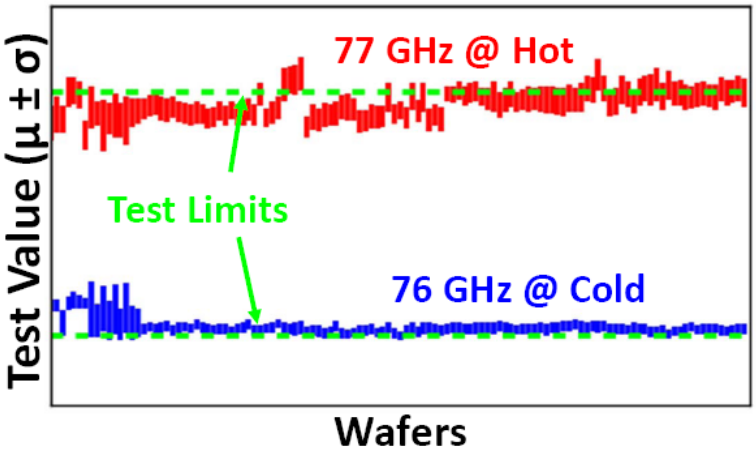


FIGURE 3.9: Yield issue due to cold/hot voltage tests

For every wafer, two vertical bars are shown, corresponding to the cold and hot results. Each bar shows the range of measured values from dies on the respective wafer. This range is $[\mu - \sigma, \mu + \sigma]$ where μ stands for the mean and σ stands for the standard deviation. The upper and lower limits are shown as two horizontal dash lines. As seen, hot values drift more frequently beyond the limit than cold values.

3.7.1 Learning a PM model

The goal is to learn a PM model from the analytics traces conducted to resolve the yield issue in Chapter 2 and apply the model to analyze the yield issue of the automotive radar chip product line.

In the earlier discussion, a *path* was described as a sequence of steps to construct a dataset. In the actual implementation, a *trace* can be a concatenation of multiple paths. For example, the first path can be used to narrow the search to a particular test and then the subsequent path would represent exploring the temporal aspect (as discussed above) based on that test.

The extension of a trace to comprise multiple paths requires defining a few additional conjoining steps. For example, the step HRY restricts the choices of y to the

selected datasets from the previous step. Similarly, the step HRX restricts the choices of x. And the step HR restricts choices of both the x-axis and y-axis. These steps contain an evaluation for the selection, for example based on a correlation value $> t$ where t is an input parameter. Finally, a general step PLOT is implemented to generate a plot.

For the result presented in this section, the input log contains 39 traces. A trace can contain 1, 2, or 3 paths. Different prefix lengths are explored and results are shown in the table below.

TABLE 3.2: Prefix length vs number of traces

Prefix length l	0	1	2	3	4	5	6
Number of traces	98990	1271	160	63	53	42	39

As the table shows, for an l-prefix rule, a larger l leads to fewer traces contained in the resulting PM model. At $l = 6$, there is no generalization. For $l \leq 1$, the numbers of traces are large, which might be considered as over generalized. One can select a perceived reasonable model to apply. For example, this can depend on the desirable runtime - a more generalized model would run slower because there are more traces to execute. In this example the model based on the 2-prefix rule was selected.

Let S_a be the set of traces in the l_a-prefix PM model, and S_b be the set of traces in the l_b-prefix PM model. Note that the property $S_a \subseteq S_b$ holds true if $l_a > l_b$. That is, the traces in a model with some prefix constraint are always all contained in models with shorter prefix constraints.

The 2-prefix PM model learned from the 39 traces is shown in Figure 3.10. Two traces are highlighted. The first trace has three paths, marked as A1, A2, and A3. This is the trace that leads to Figure 3.3. The three paths are explained below:

A1: This step uses CCA to identify that there is a high (CCA) correlation between test A and E-test PP5.

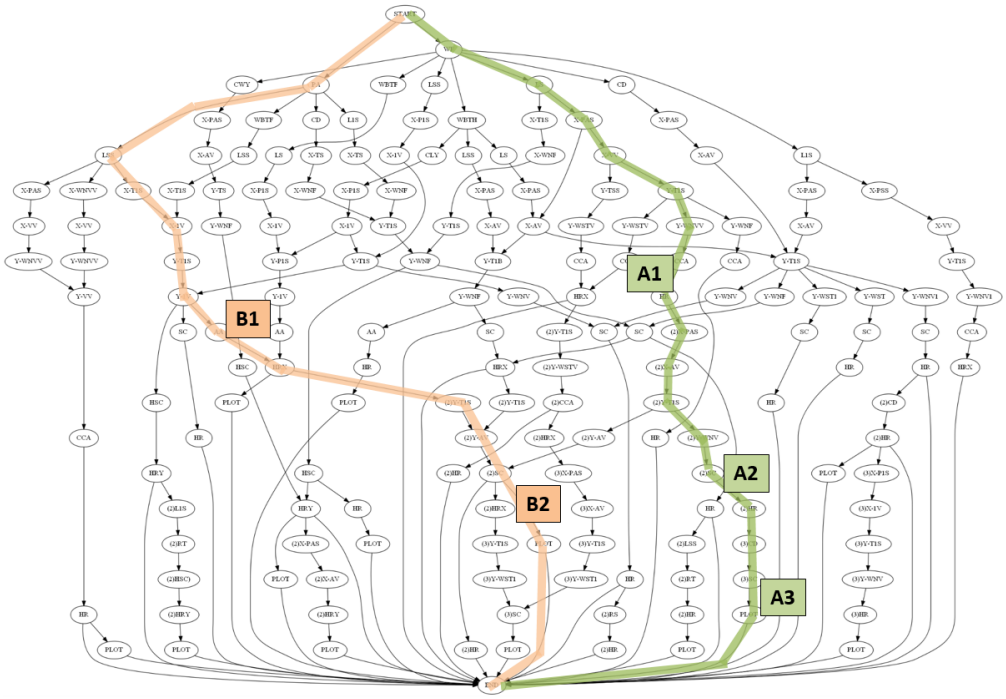


FIGURE 3.10: 2-prefix PM model learned from the 39 traces

A2: By restricting to test A, this subsequent step determines that the number of fails due to test values X_1, X_2, X_3 is highly correlated to PP5. This analysis is through standard statistical correlation

A3: This step applies a temporal consideration (i.e. step CD) to uncover the result seen in Figure 3.3.

The second trace with paths B1 and B2 is a new trace generalized by PM that was not among the 39 traces in the original input log. Though, it should be noted that this trace only exists when $prefix \leq 2$ which means that it exists among 121 new traces that had to be executed. This trace was a successful trace for analyzing the yield issue discussed in this section. The results of paths B1 and B2 are explained through the two plots in Figure 3.11.

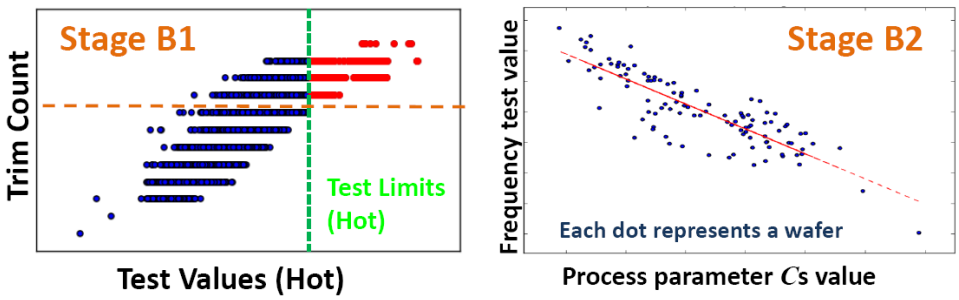


FIGURE 3.11: Finding association(left); Finding statistical correlation (right)

The left plot shows that hot pass (blue) is associated with the variable trim count. Every dot is a part. Before the cold/hot final test, the frequency of an on-chip oscillator is measured at room temperature. Then, the oscillator is tuned by a *trim* process. Trim count is treated as a test value. In stage B1, *association* analysis based on two tests is conducted. The plot shows that all dies passing the hot test have a low trim count.

Because the trim count and the frequency test are known to be associated, in the subsequent path B2, frequency test is shown to be highly correlated to an E-test C. This correlation is shown in the right plot. Note that the direct relationship between the trim count and E-test was also explored, but it did not find any meaningful result.

The trace B1→B2 is new because during analysis conducted for the work in Chapter 2, association analysis using two tests was never considered. What was considered was the association analysis shown in Figure 3.5 using two E-tests. However, the learning was able to generalize to include association analysis using two tests.

To be more specific, association analysis can be thought of as finding two lines in a 2D space as shown in Figure 3.12. In this plot, suppose red dots are failing dies and blue dots are passing dies. If one can find a vertical line such that one side of the line contains only (or mostly) one type of dots, then one can say the type of dots is associated with the x variable. Similarly, finding a horizontal line can decide if there is an association to the y variable.

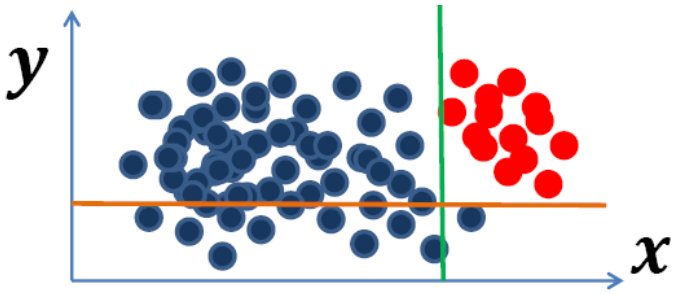


FIGURE 3.12: Association by finding two lines

In the left plot of Figure 3.11, the red type is associated with the hot test. However, this is not useful because failing dies are decided by the test. The blue type is associated with the trim count. This is useful because that indicates all passing dies have a lower trim count.

3.8 Limitations of the PM Model

While the result is encouraging, generalizing the approach to other application scenarios may require further enhancement of the current PM design.

As mentioned before, one limitation of the current PM approach is that it does not consider cross-steps dependency. Dealing with such a dependency is avoided in the PM algorithm by taking special consideration in designing the process steps. However, the consideration also constrains the design of the steps. For other applications, the PM algorithm may need to be enhanced to explicitly take cross-steps dependency into account. In addition, this PM approach does not allow loops in the process. Considering loops will drastically increase the complexity of the PM algorithm. However, it can further relax the constraint on what types of steps can be defined.

In general, there is a trade-off between the objective to simplify the PM algorithm and the objective to allow flexibility in designing the process steps. The capability of a

PM model is limited by the set of process steps. How to learn to refine and enhance a set of process steps can be another interesting future research question.

Furthermore, how applicable a PM approach is to other types of analytics applications in test remains questionable. For example, if for an application the main challenge in Figure 3.1 is not with the dataset preparation box but with the meaningfulness determination box, then further research is needed to determine if the meaningfulness determination requires a complex process or not. If the process is simple, then applying the PM approach might not make sense.

3.9 Summary

This chapter conveys the message that the analytics process is not automatic. It is shown that the result of analytics is subjective to how an analyst prepares the datasets and decides the meaningfulness of results. The proposed process mining (PM) approach was designed to learn the experience of preparing datasets from one analyst and provide that experience to another analyst, thereby reducing the subjectivity due to that step.

The presented approach is specific to learning the process of correlation analysis. A PM algorithm tailored to this application was introduced and the design of the process steps to enable learning was explained. The effectiveness of this approach was demonstrated by applying the PM model resulting from the work in Chapter 2 to resolve a yield issue in a new (and different type of) automotive product line.

Chapter 4

Generalization of an Outlier Model into a "Global" Perspective

4.1 Overview

This chapter explores the generalization of an outlier model from two perspectives, temporal and spatial. It is shown that model generalization with existing distribution-based outlier analysis methods can vary significantly. Part of this variation is shown to be due to temporal and spatial uncertainties, both of which are explained in detail. A "big data" outlier analysis approach is proposed together with a probability-based outlier evaluation for improving model generalization. Experiments are conducted based on two automotive product lines to explain the concepts and demonstrate the effectiveness of the proposed approach.

4.2 Introduction

The work in this chapter focuses on distribution-based methods, including both univariate and multivariate approaches. Recall from Section 1.1.2 and Figure 1.2 that distribution-based outlier model comprises three components:

1. A set of parts used in the analysis, called the *base set* and denoted as $\mathcal{B}$
2. A method to calculate an outlier *score* s, denoted as $\mathcal{M}$
3. A model $\mathcal{R}$ based on outlier scores to classify parts as inliers or outliers

Constructing an outlier model can be subjective. For example, in wafer-probe testing, a common practice is to let the base set $\mathcal{B}$ to comprise dies from the same wafer. This intends to find *wafer-based* outliers. Alternatively, one may expand the base set to find *lot-based* outliers.

Assume each base set is a wafer of dies. In outlier analysis, the main source of subjectivity is in the selection of the model $\mathcal{R}$. A simple experiment is presented to illustrate the impact of this subjectivity, comparing the three univariate methods: DPAT, AEC (DPAT), and RDPAT.

A DPAT model is of the form: x is an outlier iff $x \notin [\mu - k\sigma, \mu + k\sigma]$. Here, μ is the mean and σ is the standard deviation of the distribution based on all dies in $\mathcal{B}$. For each die, its test value v is converted into the outlier score $x = \frac{v}{\sigma}$ to be evaluated by the model. Therefore, determining the model is just determining the k value in the model.

An AEC model is like a DPAT model, except that the inlier range is defined differently as: $[me - 0.43 \times k(me - p_1), me + 0.43 \times k(p_{99} - me)]$, where me is the median, p_1 is value at the 1% percentile point and p_{99} is the value at the 99% percentile point of the distribution [38]. Similarly, determining the model means determining its k value.

Robust DPAT can be thought of as a hybrid of DPAT and AEC [38]. A wafer distribution is run through a process involving outlier removal [64] and a Normality test. If the distribution fails the Normality test, transformation to Normal distribution is invoked [65][66]. The resulting distribution is then checked by the Normality test. For those distributions that pass the Normality test (before or after the transformation), DPAT modeling is applied. If a transformed distribution still fails the Normality test, then AEC is applied as the last resort. For the test used in the simple experiment, all

wafer distributions pass the Normality test (before or after the transformation). Hence, one k value is determined for the RDPAT outlier model.

In the experiment, the following assumption is adopted for constructing an outlier model. Given N wafers ordered chronologically by their test time, the first 10% of the wafers are used to determine the model. Then, the model is applied to the remaining 90% of the wafers to observe its outcome. Figure 4.1 summarizes the results based on one wafer probe test for an automotive sensor product. The experiment covers 658 wafers as shown in the figure. This figure is explained below.

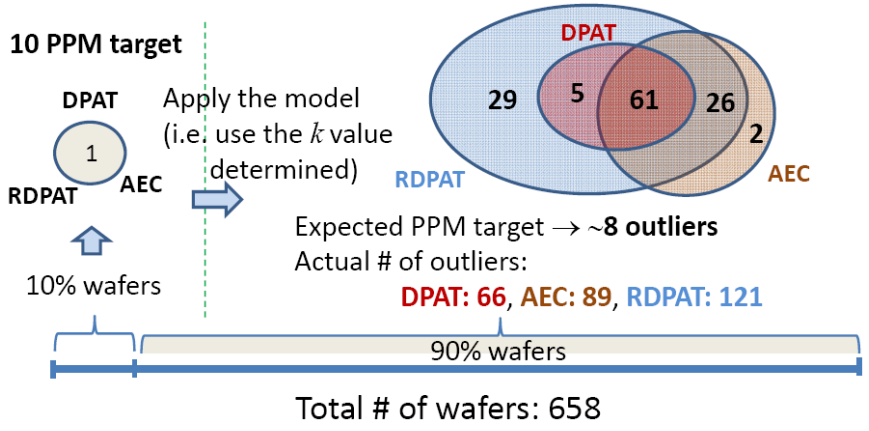


FIGURE 4.1: Comparing three univariate outlier methods

An outlier model is usually developed based on some initial qualification data. Further, an outlier model determines an outlier locally, for example usually based on only parts from the same wafer. This initial and local perspective raises an interesting question as whether an outlier property initially and locally determined can be generalized into a "global" perspective based on the future data collected over a long period of time.

To determine the k value, a PPM (part per million) yield reduction target is usually established. In the experiment, 10 PPM is assumed as the target, i.e. it is aimed at the 10 most outlying parts per million parts. The first 10% of the wafers contain roughly 96K parts. With a 10 PPM target, the k value is therefore set to screen out exactly one

part. For DPAT, this results in $k = 8.117$. For AEC, it is $k = 8.242$. And for RDPAT, the most outlying die resides at $k = 8.317$.

Figure 4.1 shows that on the 10% of the wafers, the three methods find the same part as the most outlying part. This is depicted as a single circle (with a "1" inside) to indicate the overlap of the three 1-die outlier sets.

Results of applying the models to the remaining 90% of the wafers are shown on the right. Note that these wafers contain roughly 810K parts. Hence, with 10 PPM target, the expected number of outliers is about 8 parts.

First, observe that the number of outliers screened out by each model (i.e. by using the k value determined earlier based on 10% of the wafers) deviates quite significantly from the expected number, 8. DPAT finds 66 outliers, AEC finds 89, and RDPAT finds 121. The Venn diagram then shows their overlaps. Observe that all DPAT outliers are RDPAT outliers. Moreover, the intersection of the three outlier sets has 61 outliers. Overall, the result from one method can deviate significantly from another method.

4.2.1 Multivariate outlier example

Figure 4.2 shows results from a similar experiment comparing two commonly-used multivariate methods: Mahalanobis (Mah) and Linear Regression (LR). Two highly-correlated (correlation > 0.9) tests are selected to build a 2-dimensional outlier model.

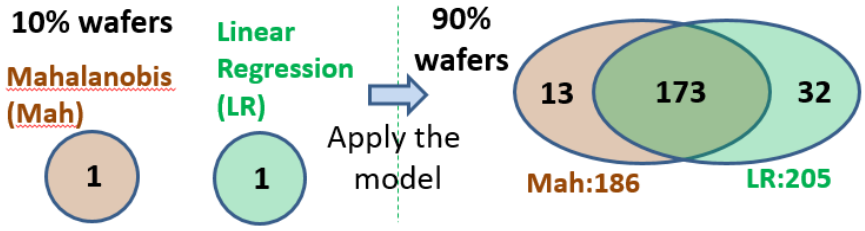


FIGURE 4.2: Comparing two multivariate outlier methods

A scikit-learn package [56] is used to estimate covariance and calculate the Mahalanobis distance. In linear regression, a regression line is estimated. Then, the distance

of each die to the line is calculated. These distances are treated as the "new measured values" for the dies. Then, DPAT is applied on the new measured values.

To show the effect only on multivariate outliers, in the experiment the two respective AEC models (one for each test) are applied first, to remove univariate outliers. Then, the two multivariate methods are applied to the remaining dies independently.

Using the first 10% of wafers, the two multivariate models find their respective most outlying die. These two most outlying dies are different as shown in Figure 4.2.

When the two multivariate models are applied to the remaining 90% of wafers, the numbers of outliers (Mah:186, LR:205) are much larger than the expected number of 8 parts. While they share 173 outliers, each model has a number of its own unique outliers as well.

The two observations with Figure 4.1 earlier apply to Figure 4.2 as well. First, the number of outliers screened out by each model deviated quite significantly from the expected number, 8. Second, the result of one method can deviate from the other method.

4.2.2 Temporal and spatial uncertainties

Figure 4.1 and Figure 4.2 reveal two uncertainties for an outlier model. The first uncertainty is exemplified with the different PPM numbers screened out by a model on the first 10% of the wafers and on the remaining 90% of the wafers. Call this *temporal uncertainty*. The second uncertainty is exemplified by the diverse results using different outlier methods. Call this *spatial uncertainty*.

Note that these two uncertainties are not necessarily independent of each other. For example, the spatial uncertainty makes it harder to justify the excessive yield loss seen in the above examples, due to the temporal uncertainty.

The two uncertainties motivate a search for an outlier analysis approach that improves on the existing methods by satisfying two properties:

1. The outcome of outlier screening is more robust over the production time (i.e. less temporal uncertainty)
2. The outliers screened out are more agreeable among the methods (i.e. less spatial uncertainty)

The term *model generalization* is used to denote such properties because the outcome from such a model would be more *generalizable* over time and across methods.

4.3 Understanding the uncertainties

In this work, model development is done in a proactive fashion by assuming a PPM target. Ideally, one would like to screen out only those dies that are defective. Hence, instead of starting with a PPM target, a common practice could to conduct the the model development in a reactive fashion by using a set of known defective parts based in some qualification lots. For example, these parts can be parts that fail in a burn-in experiment [11] or parts returned by the customer [8].

However, starting with qualification lots with known defective parts does not fundamentally change the issues with the two uncertainties discussed above. When a model is applied in future production, the outcome of the model can still deviate significantly from that observed on the qualification lots.

There are additional challenges that present themselves in practice. For example, it is challenging to ensure that the set of known defective parts is sufficient to represent the underlying defect universe of interest. Moreover, determining if there exists an outlier model to screen a given defective part could also be challenging as well [16]. Compounding to this challenge is the fact that outlier model existence also depends on the allowable PPM loss.

In some scenarios, it is preferable to not wait until known defective parts become available. For example, for customer return prevention, it is undesirable to wait for a

customer return. In such a scenario, one would like to develop outlier models without customer returns.

For these reasons, this work simply assumes a PPM target to start the model development. This allows the study to focus on the two uncertainties highlighted above.

4.3.1 Further illustration of temporal uncertainty

The temporal uncertainty observed in Figure 4.1 is not unique to that particular test. Figure 4.3 shows results of running similar experiments across 756 wafer probe tests on the sensor product. In these experiments, the same 10 PPM budget and the same strategy of 10%-90% split of total wafers are used. In the plots, each (blue) dot represents a test.

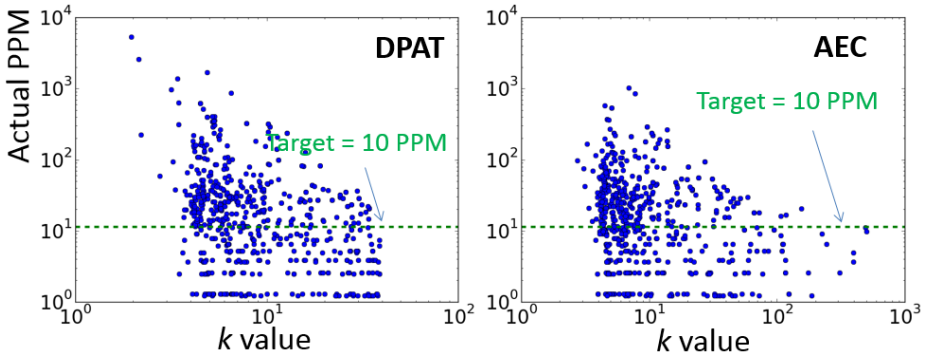


FIGURE 4.3: Temporal uncertainty across 756 tests

Figure 4.3 shows that the temporal uncertainty exists for many tests, and for both DPAT and AEC, i.e. the actual PPM deviates significantly from the target 10 PPM. Moreover, this uncertainty has almost no correlation to the k value.

Table 4.1 then shows the results of repeating the experiments by varying the PPM target. With each PPM target and each method, the table reports two numbers: the $mean$ which is the average of actual PPM numbers obtained for all tests, and the std which is their standard deviation.

TABLE 4.1: Actual PPM for a PPM target across all tests

PPM	DPAT		AEC		RDPAT	
Target	*mean*	*std*	*mean*	*std*	*mean*	*std*
10	51.5	244.2	32.4	75.9	153.1	561.4
20	94.2	449.3	54.5	96.5	226.9	767.7
30	120.4	480.7	72.6	109.1	280.7	842.7
40	150.2	516.2	88.4	118.5	326.8	940.7
50	185.3	581.4	106.1	130.5	377.5	1058.4
60	220.9	819.4	121.6	138.4	427.3	1241.3
70	247.2	934.9	136.4	146.0	464.5	1347.2
80	271.8	978.1	150.8	154.1	494.8	1390.0
90	297.7	1039.5	164.6	161.2	528.9	1461.3
100	317.6	1064.2	178.6	170.3	556.3	1501.4

Table 4.1 essentially shows that temporal uncertainty is not unique to the 10 PPM target used in the earlier experiments. As the PPM target increases, both the $mean$ and std increase. The table shows that AEC is more immune to temporal uncertainty, i.e. it consistently has smaller numbers compared to the other two methods. However, even with PPM target 100, the AEC method results in $mean + std = 178.6 + 170.3 = 348.9$ PPM, which means there are tests whose actual PPM numbers are more than three times larger than the PPM target. In fact, there are 74 such AEC models (from 74 tests).

4.3.2 Further illustration of spatial uncertainty

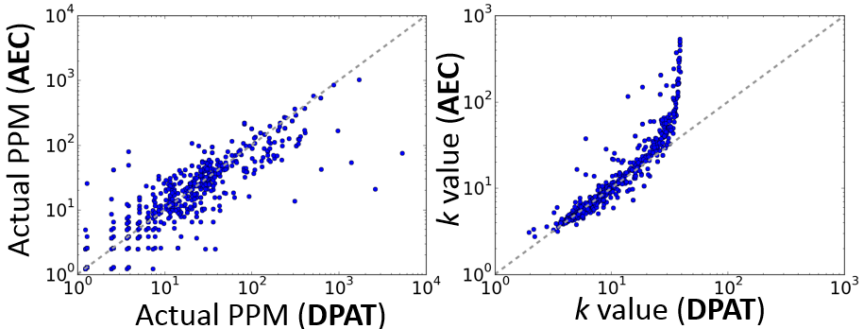


FIGURE 4.4: Spatial uncertainty across 756 tests

Figure 4.4 then illustrates the spatial uncertainty between DPAT and AEC across all the tests. Again in these plots, each (blue) dot is a test. On the left plot, observe that on many tests the number of DPAT outliers is quite different from the number of AEC outliers. On the right plot, observe that their k values can also differ quite significantly, especially for DPAT models with a large k value.

4.3.3 Analyzing the result in Figure 4.1

The above results show that temporal and spatial uncertainties are not unique to the one test used to produce the result shown in Figure 4.1. The result in Figure 4.1 is analyzed in more detail below.

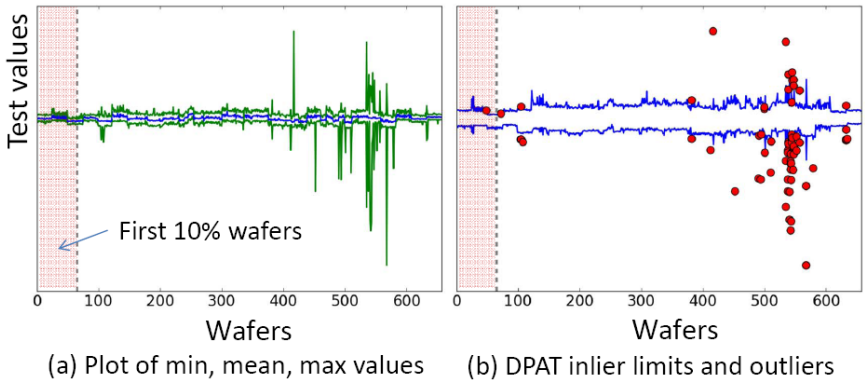
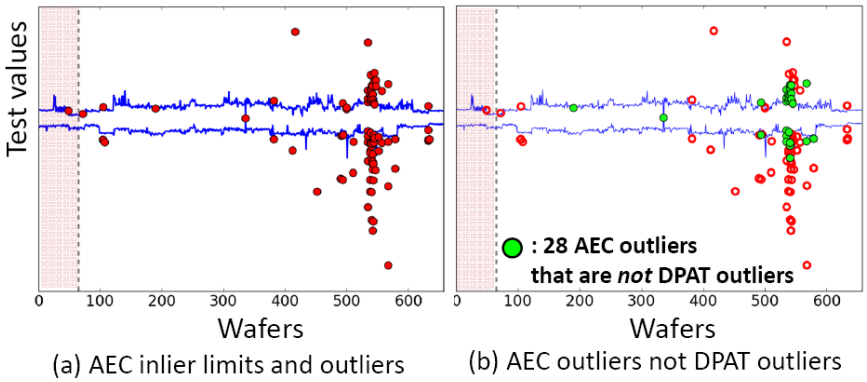


FIGURE 4.5: Temporal fluctuation and its impact to DPAT limits

Figure 4.5-(a) plots the mean, min, and max measured test values for each wafer across the 658 wafers. The first 10% of the wafers are highlighted in the plot. Observe that after 400 wafers there are significant fluctuations of the min/max values. It therefore makes sense that because the k value of an outlier model is determined using the first 10% wafers, the resulting model does not account for the much larger test value fluctuations seen later in production.

To confirm this conjecture, Figure 4.5-(b) shows the upper and lower limits set by the DPAT model on each wafer (recall $k = 8.117$). The DPAT outliers are shown as

red dots. Observe that the outliers show some concentration on the area with large fluctuation. Figure 4.6-(a) shows a similar plot using the AEC limits and their resulting outliers.



(a) AEC inlier limits and outliers (b) AEC outliers not DPAT outliers

FIGURE 4.6: AEC outliers, and comparing to DPAT outliers

From these plots, it can be seen that temporal uncertainty is due to the significant change of statistics between the first 10% wafers and the remaining 90% wafers. In other words, the first 10% wafers are not representative enough to capture the statistics seen on later wafers.

Figure 4.1 earlier shows that 28 AEC outliers are not DPAT outliers. Figure 4.6-(b) highlights where those 28 AEC-unique outliers (green dots) are located. It is interesting to observe that these 28 AEC outliers tend to be "marginal" where most of them are close to the AEC limits.

Figure 4.6-(b) reveals that the divergence seen in the Venn diagram in Figure 4.1 might be largely due to the fact that different models treat "marginal" outliers differently. If that is the case, then the divergence (i.e. spatial uncertainty) can be reduced by focusing on finding only the "gross" outliers.

4.4 Identifying "gross" outliers

To confirm the conjecture that spatial uncertainty mostly occurs on marginal outliers, a method is needed to differentiate marginal outliers from gross outliers. For this purpose, a so-called *marginality test* is developed. The desired functionality of this test is such that if an outlier passes the marginality test, it is classified as a gross outlier. Otherwise, it is deemed as a marginal outlier.

4.4.1 The concept of marginality test

Recall that an outlier is a die that has its outlier score s falling beyond the inlier range $[L, U]$ where L is the lower limit and U is the upper limit. Different methods calculate the outlier score and the inlier range differently. Regardless of how they are calculated, the calculations are based on the parts in the base set $\mathcal{B}$.

In the experiments above, the base set contains dies from the same wafer. Hence, an outlier is found relatively to other parts in the same wafer. To test if the outlier is marginal, the idea is to move the outlier into another *similar* wafer. If the outlier becomes an inlier on a similar wafer using its respective inlier range, then the outlier is marginal. In other words, a marginal outlier is defined as an outlier where there *exists a similar wafer* that treats the part as an inlier.

4.4.2 Using the N most similar wafers

Suppose the marginality test is applied using the N most similar wafers. This requires a definition of *wafer similarity*. In this work, wafer similarity is defined by measuring the similarity between two wafer distributions.

There are many choices of measurements for the similarity between two probability distributions, such as Kolmogorov-Smirnov (KS) statistic, differential entropy, or using moments as features. The work in this chapter simply encodes a distribution by its

mean and standard deviation as a 2-dimensional vector. Then, given two vectors, their Euclidean distance is used as the inverse measure of wafer similarity. To avoid bias, both mean and standard deviation values are normalized to a value in range $[0, 1]$.

For the experiments reported, this simple encoding works quite well in most cases. Hence, optimization of the wafer similarity measure was left to future work.

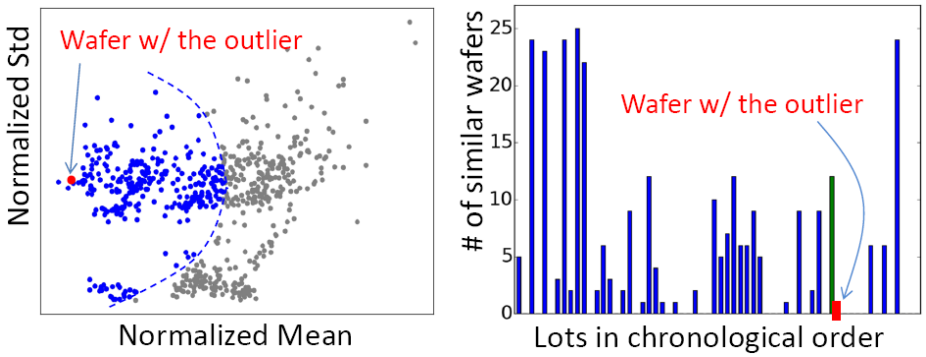


FIGURE 4.7: Illustration of 300 most similar wafers to the wafer containing a marginal outlier

Figure 4.7 shows an example wafer and its 300 most similar wafers. On the left, the blue dots are the similar wafers under consideration, while the gray dots are the remaining wafers. Each dot represent the wafer distribution positioned by its normalized mean (x-axis) and standard deviation (y-axis). The right plot shows where those 300 similar wafers are located chronologically with respect to the example wafer. The x-axis shows the lot indices arranged chronologically by test time. The y-axis shows the number of similar wafers in the lot.

4.4.3 Examples of marginality test

Figure 4.8 shows a marginal outlier example and a gross outlier example (i.e. not marginal). The marginal outlier example is the same example shown in Figure 4.7. The two examples in Figure 4.8 are based on the DPAT model. Recall that the DPAT model has the k value $= 8.117$, i.e. the inlier range is $[\mu - 8.117\sigma, \mu + 8.117\sigma]$. The

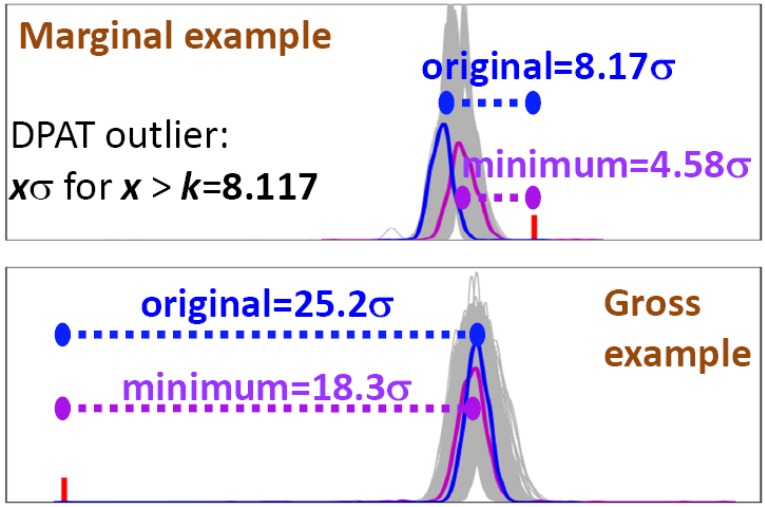


FIGURE 4.8: Examples of marginal and gross outliers

x-axis is the measured test value where both plots are shown with the same scale. The location of the outlier is marked by a short vertical red bar.

For the marginal outlier, its outlier score is 8.17 (the test value $= 8.17\sigma$ where the σ is specific to the wafer). Hence, the part is classified as an outlier based on the wafer distribution (blue). Then, this outlier is tested against 300 most similar wafers. The distributions of these 300 similar wafers are plotted using gray color to show their overall span in the background.

With the 300 wafers, each wafer gives a new outlier score for the part using its respective σ. Among them, the minimum is 4.58 as illustrated in the plot. The wafer distribution producing this minimum is plotted in purple. Because $4.58 < k = 8.117$, the outlier is deemed marginal based on the earlier marginality definition.

In contrast, the second plot shows a gross outlier example. Among the 300 most similar wafers, the minimum outlier score is 18.3 which is much larger than the k value 8.117. Hence, this outlier is not marginal. Comparing the two plots it can be seen pictorially that the marginal and gross outliers meet our intuition — the marginal outlier is much closer to the distributions than the gross outlier.

4.4.4 Gross outliers in view of Figure 4.1

The marginality test is implemented based on finding the 300 most similar wafers. Then, it is applied with the DPAT, AEC, and RDPAT models independently to find their respective gross outliers.

Figure 4.9 shows the resulting gross outliers for each model. The numbers of gross outliers found for DPAT, AEC and RDPAT models are 33, 48 and 74, respectively. They are shown in three separate Venn diagrams in view of the the original Venn diagram in Figure 4.1.

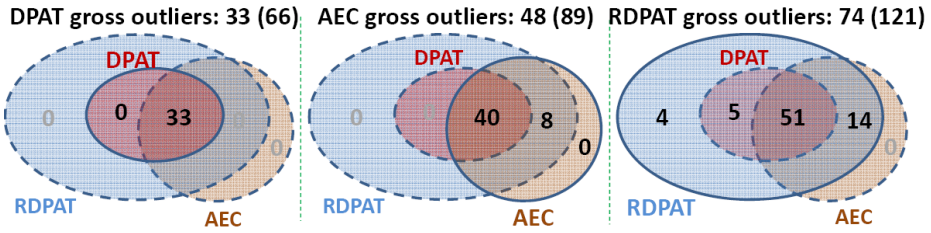


FIGURE 4.9: Gross outliers in view of the Venn diagram shown in Figure 4.1. A(B): A is the number of gross outliers and B is the number of outliers copied from Figure 4.1.

Notice in Figure 4.9 that the 33 DPAT gross outliers are all inside the intersection of the *original* three outlier sets. In other words, if one could have a model that finds only those 33 gross outliers, the spatial uncertainty issue discussed earlier would have been resolved because every outlier found by this model would be agreeable by the original three PAT models as well.

Improvement to the AEC outlier set is less noticeable. However, notice that the two AEC-unique outliers shown in Figure 4.1 are no longer present in Figure 4.9 (a "0" is shown). In other words, those two AEC-unique outliers are marginal and get removed by the marginality test. Consequently, spatial uncertainty is reduced.

There is no obvious improvement to RDPAT. The RDPAT gross outliers still spread across different regions of the Venn diagram as in Figure 4.1, except the numbers in

undesirable regions are relatively smaller.

4.4.5 Gross outliers vs. marginal outliers

Let the outliers in the *union* of the three gross outlier sets be called the *gross outliers*. Hence, a gross outlier is an outlier defined to be gross with respect to *any* of the three methods. Let the outliers in the *intersection* of the three gross outlier sets be called the *shared gross outliers*. Note that there are 32 shared gross outliers (1 DPAT gross outlier is not shared). Let any outlier that is not a gross outlier be called a *marginal outlier*.

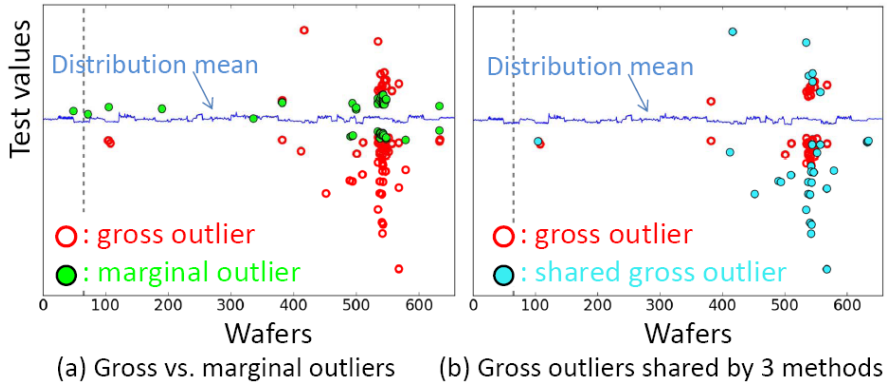


FIGURE 4.10: Marginal vs. gross vs. shared gross outliers

Figure 4.10 then shows where those different types of outliers are located using a similar illustration to that in Figure 4.6. On the left plot, marginal outliers are seen to reside closer to the mean of the wafer distribution than the gross outliers. Pictorially this matches our intuition for what a marginal outlier should be. Interestingly on the right plot, a similar situation is observed between shared gross outliers and gross outliers (that are not shared). Shared gross outliers tend to be further away from the wafer distribution mean.

The above results show that the proposed marginality test concept is reasonable and can be used to differentiate marginal outliers from gross outliers in an intuitive sense. Moreover, spatial uncertainty can be reduced as different models tend to be

more agreeable on gross outliers. Building on these observations, a new outlier analysis approach is devised in section 4.5.

4.5 The proposed outlier analysis approach

The idea of a marginality test can be incorporated into an outlier analysis approach to find gross outliers. Such an approach can potentially provide two benefits:

(1) By avoiding marginal outliers, excessive yield loss is reduced. This in turn reduces temporal uncertainty. For instance, in the example in Figure 4.1, DPAT originally finds 66 outliers (with respect to the expected number 8). After the marginality test, 33 gross outliers remain. In a sense, the temporal uncertainty is reduced from $\frac{66}{8} = 8.25$ (8.25 times over the expected target) to $\frac{33}{8} = 4.125$.

(2) Gross outliers are more likely to be agreeable among various methods, and hence, focusing on finding gross outliers can help reduce or even remove spatial uncertainty.

4.5.1 A "big data" perspective

To incorporate marginality test, a new outlier analysis approach is proposed, which follows a "big data" perspective. The term *big data* is used to emphasize that the outlying property of a given part is evaluated using all wafer data available prior to the wafer containing the part. Hence, the data in use is accumulative. Figure 4.11 illustrates this perspective.

As illustrated in the figure, the approach comprises three components:

(1) A component to decide potential outliers on a given wafer w_i which should be evaluated further

(2) A component to decide N most similar wafers to the wafer w_i

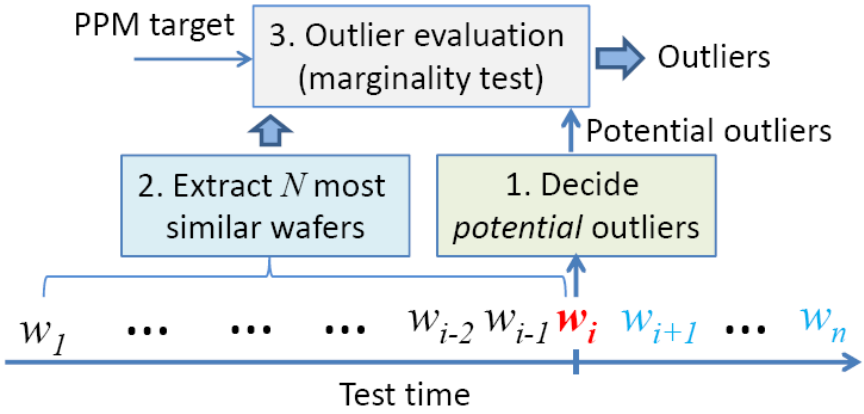


FIGURE 4.11: Outlier analysis following a big data perspective

(3) A component to perform outlier evaluation using the marginality test idea discussed above

4.5.2 Issue with using a DPAT or AEC model

If Figure 4.11 were to be implemented with a DPAT or AEC model, the model would be used in component (1) to decide the potential outliers. This is how it is used with the marginality test experiments presented earlier in Section 4.4. Figure 4.12 illustrates this use in component (1).

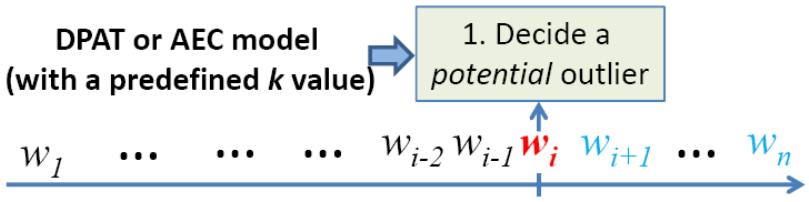


FIGURE 4.12: Potential outliers decided by a DPAT or AEC model

The issue with using a DPAT or AEC model alike is that the k value (i.e. the outlier boundary) is predefined. For the one example discussed earlier this would not be a problem because the problem there is that the k value is too small, resulting in more outliers than desired. On the contrary, if the predefined k value is too large, then a

gross outlier on a later wafer would not have been selected into the potential outlier set. Consequently, such a gross outlier would have no chance to enter the outlier evaluation component.

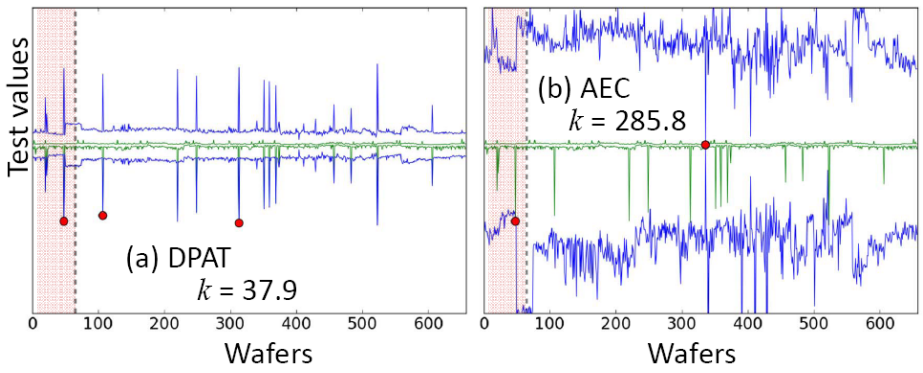


FIGURE 4.13: Under-screen example for PPM target = 10

Figure 4.13 exemplifies such a situation. The plots show the min/max measured values (green) as well as the upper and lower limits (blue) of the outlier models. The outliers found are shown as red dots. On the left, the DPAT model finds 2 outliers in the remaining 90% wafers. On the right, the AEC model finds only 1 outlier. In a sense the k value decided based on the first 10% wafer is too large. From the min/max values (green), one can infer that there can more potential outliers could be extracted.

4.5.3 Adaptive k value and its potential issue

One potential way to fix the issue above is to adaptively adjust the the k value. Figure 4.14 illustrates this process.

For a PPM target of 10, on average one most outlying die is supposed to be identified every 100K dies. Intuitively, after ever 100K dies the k value could be recomputed as shown in Figure 4.14. While this might alleviate the under-screen problem, it does not resolve it. For example, suppose there is a gross outlier in w_i. Whether or not this

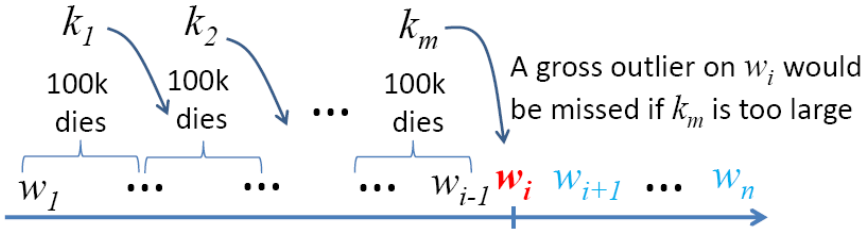


FIGURE 4.14: Issue with adaptive k value (10 PPM target)

die can be selected into the potential outlier set for further evaluation depends on the previous k_m, which can still be too large.

The idea of adapting k becomes less effective when the PPM target is even smaller. For example, for a 1 PPM target, k would be adjusted every 1M dies. This may be too infrequent to capture the change of statistics in the production data.

4.6 Probability-based outlier evaluation

This section first describes the implementation of the outlier evaluation component, then continues to describe how potential outliers are decided in light of the implementation.

Assume that a PPM target t is given. For example, for 10 PPM, $t = 10^{-5}$. To evaluate a potential outlier die, calculate its probability of occurrence. For a measured test value v, this probability is either $p(x \geq v)$ if v is on the right side of the distribution, or $p(x \leq v)$ if v is on the left side. Here $p()$ is the probability density function estimated based on the set of dies, i.e. the base set.

4.6.1 Estimating probability of occurrence

Kernel density estimation (KDE) is a popular approach for estimating the probability density function of a set of points [67][20]. Figure 4.15 illustrates its basic concept.

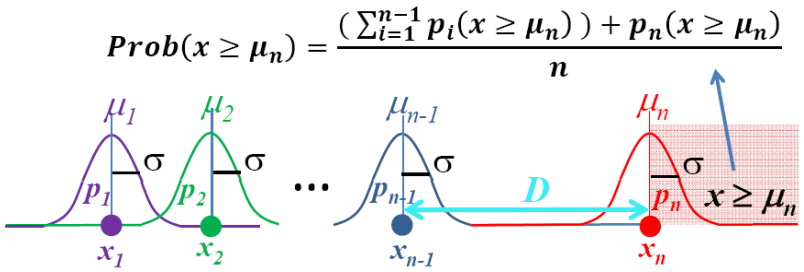


FIGURE 4.15: Illustration of kernel density estimation (KDE)

Suppose there are n dies $x_1, \ldots, x_n$ with measured test values $\mu_1, \ldots, \mu_n$. Suppose x_n is a potential outlier. In KDE, each die x_i is associated with a Normal distribution $p_i = N(\mu_i, \sigma)$. Figure 4.15 then shows how the probability of occurrence for x_n is calculated as $Prob(x \geq \mu_n)$. Note that $p_n(x \geq \mu_n) = \frac{1}{2}$. In KDE, the *bandwidth* σ is estimated based on the n points in the set. Bandwidth estimation for the experiments in this chapter is performed using the popular Scott & Silverman method [67].

4.6.2 Heuristic for fast probability estimate

KDE calculations illustrated in Figure 4.15 can be expensive: $n-1$ Normal distributions need to be evaluated. A heuristic is therefore used to speed up the evaluation.

Suppose the PPM target is $t = 10^{-5}$. In Figure 4.15, for x_n to be an outlier, its probability needs to be $Prob(x \geq \mu_n) < 10^{-5}$. Suppose the distance between x_{n-1} and x_n is D and $D > 4.754\sigma$. Then, based on the Normal distribution for $p_{n-1} = N(\mu_{n-1}, \sigma)$, it follows that $p_{n-1}(x \geq \mu_n) < 10^{-6}$. Because the distances of $x_1, \ldots, x_{n-2}$ to x_n are all larger than D, it also follows that $\forall i, p_i(x \geq \mu_n) < 10^{-6}$. As a result the term "$\sum_{i=1}^{n-1} p_i(x \geq \mu_n)$" would be $< (n-1) * 10^{-6}$.

To get $Prob(x \geq \mu_n) < 10^{-5}$, n has to be:

$$\frac{(n-1) * 10^{-6} + 0.5}{n} < 10^{-5} \Rightarrow n > \frac{5}{9} * 10^5 \tag{4.1}$$

In other words, verifying that $D > 4.754\sigma$ with $n \geq 55556$ dies implies that $Prob(x \geq \mu_n) < 10^{-5}$ under a normality assumption.

4.6.3 Probability-based marginality test

The above heuristic provides a convenient way to merge probability estimation into marginality test. Suppose N wafers $w_1, \ldots, w_N$ are given where w_N contains one potential outlier o to be evaluated. Suppose on wafer w_i, the closest die to o has a distance D_i. Then, the evaluation can proceed by simply checking that $D_i > 4.754\sigma$ (for PPM target $t = 10^{-5}$) for all $i = 1, \ldots, N$.

Further, ensuring that the N wafers contain more than, say 55556 dies, indicates that the probability of occurrence is $< 10^{-5}$. Hence, as long as N is large enough, the check would find outliers that simultaneously meet the PPM constraint and pass the marginality test.

Without the fast heuristic, alternatively suppose the term "$\sum_{i=1}^{n-1} p_i(x \geq \mu_n)$" is calculated by letting $p_i(x \geq \mu_n) = 0$ for all $p_i(x \geq \mu_n) < 10^{-9}$. This would speed up the computation. However, a complete experiment run for all tests would still take more than a week to finish for the data from the the sensor product used for this chapter. In contrast, with the fast heuristic, the same run would take about 1 day.

4.6.4 Evaluating multiple potential outliers

One caveat with using the heuristic arises when there are multiple potential outliers to be evaluated together. For example, in Figure 4.15, suppose there is another potential outlier x_{n+1} residing on the right and close to x_n. In this case, equation (4.1) needs to be modified.

It is intuitive that the modification can simply be changing the "0.5" in equation (4.1) to "1.5". As a result, a larger n would be required with $n > \frac{15}{9} * 10^5$, which means

about 167K dies would be needed. In general, to consider k potential outliers together, $n > \frac{10*(k-1)+5}{9} * 10^5$ dies are needed for the PPM target $t = 10^{-5}$.

4.6.5 Deciding potential outliers

Suppose the distance constraint D has been calculated (as shown in Figure 4.15 and explained above). Based on the heuristic discussed above, Figure 4.16 illustrates how potential outliers can be decided. On each tail of a wafer distribution, the objective is to scan inward to find the first "gap" where the distance between two dies is $> D$. For example, in Figure 4.16 the first gap is found with D_1. Then, the dies o_3, o_4 are treated as the first group to be evaluated. If o_3, o_4 are deemed as outliers, then scan proceeds to the next gap, in this case D_2. As a result, o_1, o_2 are evaluated. The process stops when either the dies evaluate to be inliers or no more gaps can be found.

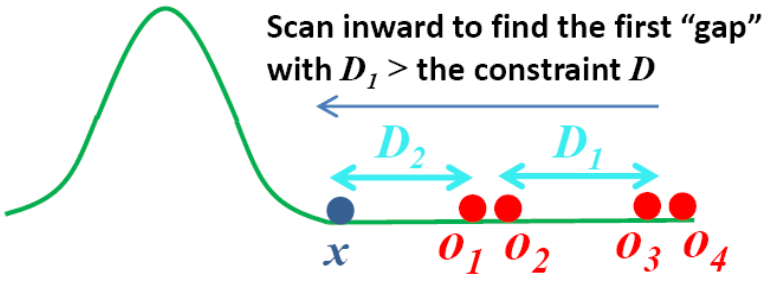


FIGURE 4.16: Deciding potential outliers

4.7 Probability-based online outlier evaluation

For the experiment, suppose there are m wafers $W = \{w_1, \ldots, w_m\}$ ordered chronologically by their test time. Figure 4.17 explains the setup for the experiment where wafers are processed following the ordering. This scheme is called *online evaluation* because when processing w_i, future wafers $w_{i+1}, \ldots, w_m$ are assumed unavailable.

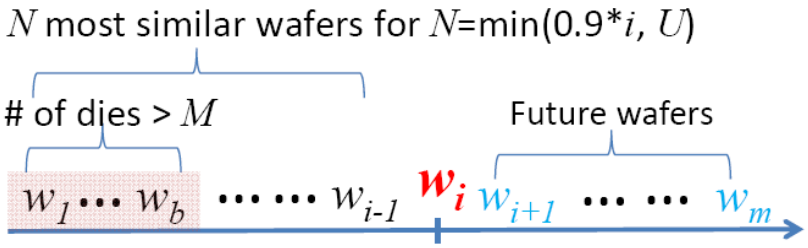


FIGURE 4.17: Setup for online outlier evaluation

When w_i is processed, N most similar wafers from among the historical wafers $w_1, \ldots, w_{i-1}$ are found. Let $N = \min(0.9 * i, U)$. The term "$0.9 * i$" means that no more than 90% of the historical wafers would be used. This is to exclude the situation where very dissimilar wafers are used. The term U is an upper bound on how many wafers to use. U is set to half the size of W but no more than 2K due to run time consideration.

For a PPM target t, a minimum constraint M is calculated such that the first b wafers contain just enough dies to have more than M. For example, the above discussion explains that for $t = 10^{-5}$, at least 55556 dies are needed. With the constraint on N using up to 90% of the wafers, enough wafers are needed so that 90% of them contain more than 55556 dies. For the experiments with the sensor product, setting $b = 50$ wafers (2 lots) would be more than sufficient.

When an outlier is found on w_i, it is removed so that when w_i is selected in the future as one of the most similar wafers, the outlier will not have an impact.

4.7.1 Handling the first b wafers

The first b wafers are handled separately. These b wafers are assumed to be given together. First, all dies in these wafers are ranked based on their test values. Then, working from the largest value, each time the most outlying die is checked by the outlier evaluation. The evaluation uses 90% of the most similar wafers among the b wafers. If the die is deemed an outlier, it is removed, and the next most outlying die is

considered. This continues until a die fails the evaluation. The process also repeats by working from the smallest value to find outliers on the opposite side of the distribution.

After the two processes, outliers are removed from the first b wafers. This is done before the online evaluation starts from wafer w_{b+1}.

4.7.2 Online outlier vs. Global outlier

The outliers found by the online evaluation above are called *online outliers*. Suppose the run finishes to the last wafer w_m. Also suppose all outliers are removed from their wafers. Then, for each outlier, outlier evaluation is re-applied by finding the N most similar wafers from the *entire* wafer set W. This use of the entire wafer set is parallel to the marginality test discussed earlier in Section 4.4.2. The goal here is to check if an online outlier would still be an outlier if the evaluation were allowed to include the *future wafers*. If an outlier passes this check, it is called a *global outlier*.

4.7.3 Comparison to earlier results

The probability-based outlier approach is applied to the test used to produce Figure 4.1 and its subsequent results, and also to the under-screen test example from Section 4.5.2.

The result is first compared to the result in Figure 4.10-(b). Recall that there are 32 *shared gross outliers*, representing the best result in the discussion before. In contrast, the probability-based approach finds 13 *online outliers*. Figure 4.18-(a) shows the locations of these 13 online outliers and the remaining $32 - 13 = 19$ shared gross outliers where all 13 online outliers are also shared gross outliers. The upper and lower limits (blue) given by the probability method are also shown. Recall that $b = 50$ wafers. Hence, the region of the first 50 wafers is highlighted in pink.

Pictorially, observe that those 19 remaining outliers tend to be more marginal than the online outliers (and recall from the earlier discussion that other DPAT/AEC /RD-PAT outliers tend to be more marginal than the shared gross outliers).

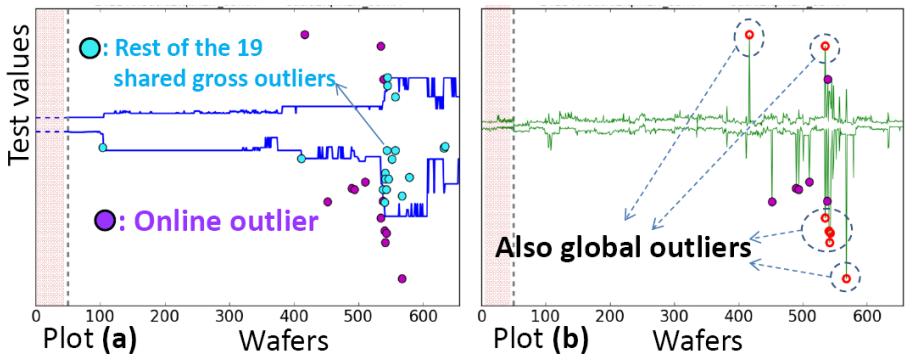


FIGURE 4.18: Comparison to result in Figure 4.10-(b)

Plot (b) shows the locations of 7 global outliers out of the 13 online outliers. The background shows the min/max values (green) across wafers. It is interesting to note that the 7 global outliers are the most outlying dies (5 on one side and 2 on the other side), if all dies from all wafers had simply been ranked using their measured test values.

Given all wafers at once and asked to find 7 outliers, the 7 global outliers would be the ideal answer. In practice, achieving this exact answer is hard because wafers become available sequentially, and an outlier decision has to be made at each wafer appearance (in an online way).

The 13 online outliers are contained in the 32 shared gross outliers which are contained in the 61 shared outliers (i.e. the intersection in Figure 4.1). Hence, the set of the online outliers would be the best with respect to temporal uncertainty.

For spatial uncertainty, the 13 online outliers are inside the intersection of DPAT, AEC, RDPAT gross outlier sets, and hence, have no spatial uncertainty in that regard.

Next, the probability-based approach is applied to the under-screen example and results are compared to Figure 4.13-(a). The probability-based approach finds 16 online

outliers shown in Figure 4.19-(a) (with upper/lower limits (blue)). In comparison, the DPAT model originally finds only 3.

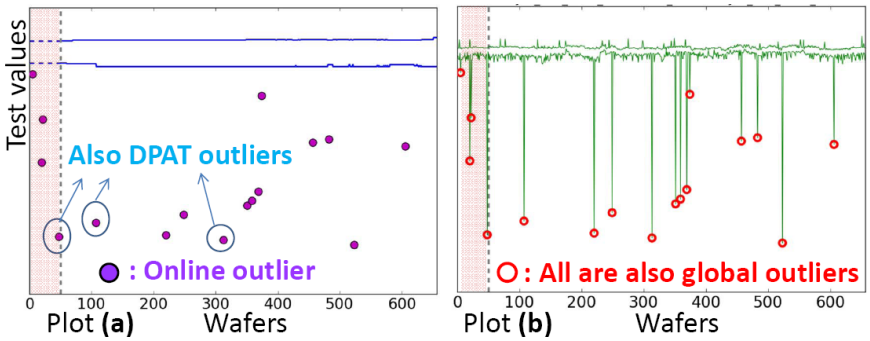


FIGURE 4.19: Comparison to result in Figure 4.13-(a)

All 16 online outliers are also global outliers. Plot (b) shows their locations in the min/max value (green) plot. It is interesting to note again that these 16 outliers were the most outlying dies if all dies from all wafers had been ranked using their test values. Pictorially, except for the 1st outlier, the other 15 outliers deviate significantly from the trend. Hence, even if the target is to screen 9 outliers in total, one would not consider screening those 15 an unreasonable result.

The results above show that for the over-screen example (excessive yield loss), the presented approach finds much fewer outliers by focusing on finding the truly gross outliers. For the under-screen example, the approach finds more outliers by finding outliers deviating significantly from the overall trend. Both are desired properties for outlier screening.

Figure 4.19 suggests that a PPM target should be used as a guide rather as a strict constraint. If there are many dies that deviate significantly from the trend, going over the target might be more reasonable than trying to put an inlier/outlier cut between two grossly outlying dies.

4.8 Comprehensive experimental results

The proposed outlier approach is applied to the 756 tests from the sensor product. The intersection of the two outlier sets found by DPAT and AEC models is used for comparison. For simplicity, these are called the "**I**-outliers."

Based on the discussion above, the results are separated into two categories: (1) The first comprises tests where the number of **I**-outliers > 9 (based on the 10 PPM target). Call this the *over-screen* category. (2) The second comprises the remaining tests where the number of the **I**-outliers ≤ 9. Call this the *under-screen* category.

The symbol **I** is used to denote the set of **I**-outliers. Symbols **O** and **G** are used to denote the sets of online outliers and global outliers, respectively (see Section 4.7.2).

Figure 4.20 shows the results of the 306 tests in the over-screen category. For this category, the comparison focuses on the number of **I**-outliers ($|\mathbf{I}|$), the number of online outliers ($|\mathbf{O}|$), and their intersection ($|\mathbf{I} \cap \mathbf{O}|$).

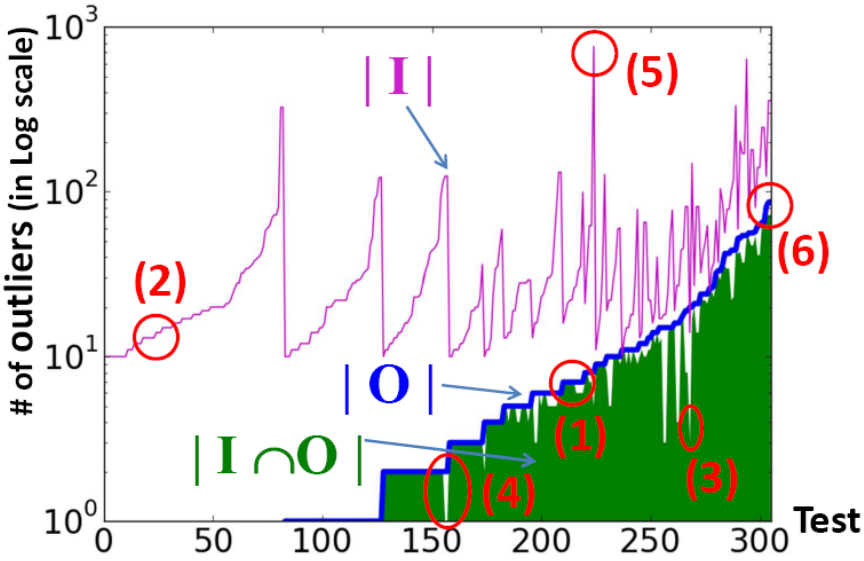


FIGURE 4.20: Result summary for over-screen cases

Observe that: (1) the number $|\mathbf{O}|$ is consistently smaller than $|\mathbf{I}|$, indicating reduced temporal uncertainty; (2) the intersection $|\mathbf{I} \cap \mathbf{O}|$ is close to $|\mathbf{O}|$ for most cases. In fact,

an **I**-coverage calculated as $100 * \frac{|\mathbf{I} \cap \mathbf{O}|}{|\mathbf{O}|}$ results in an average value of 90.95% across the tests (excluding tests having zero online outliers). This means that most of the online outliers are also I-outliers, i.e. low spatial uncertainty.

Six examples (tests) are selected from Figure 4.20 for further illustration. Results are shown in Figure 4.21. In each plot, the min/max measured values (green) on each wafer are shown (similar to other plots shown before). Different types of outliers are shown in different colors. Outliers common to both **I**-outliers and online outliers are shown in red. Outliers unique to **I**-outliers are shown in light blue. Outliers unique to online outliers are shown in purple.

Example (1) shows a case with 22 I-outliers and 7 online outliers. All the 7 online outliers are also I-outliers and hence, the **I**-coverage is 100% ("$\mathbf{I} \cap \mathbf{O} = \mathbf{O}$:7"). Observe that the remaining $22 - 7 = 15$ I-outliers are all more marginal than the 7 shared outliers. In this case, it can be said that the online outliers exhibit less temporal uncertainty, zero spatial uncertainty, and make more intuitive sense.

Example (2) shows a case with zero online outliers. Even though there are 16 I-outliers, pictorially they tend to be marginal outliers. In this case, it can also be seen that having zero outliers makes more sense.

Examples (3) and (4) show two cases where the **I**-coverage is not 100%. In example (3) $|\mathbf{I}| < |\mathbf{O}|$, and in example (4) $|\mathbf{I}| \gg |\mathbf{O}|$.

For example (3), notice that there are several online outliers right before the wafer index 100. Those outliers are missed by both the DPAT and AEC models (not I-outliers). Also, there are two online outliers above the trend line which are not I-outliers. Moreover, those 11 I-outliers not covered by the online outliers tend to be visually more marginal.

For example (4), there are only 2 online outliers which are also the most outlying dies. One of them is an **I**-outlier but the other is not. Hence, the DPAT and AEC models capture 124 outliers but miss one of the two obvious ones.

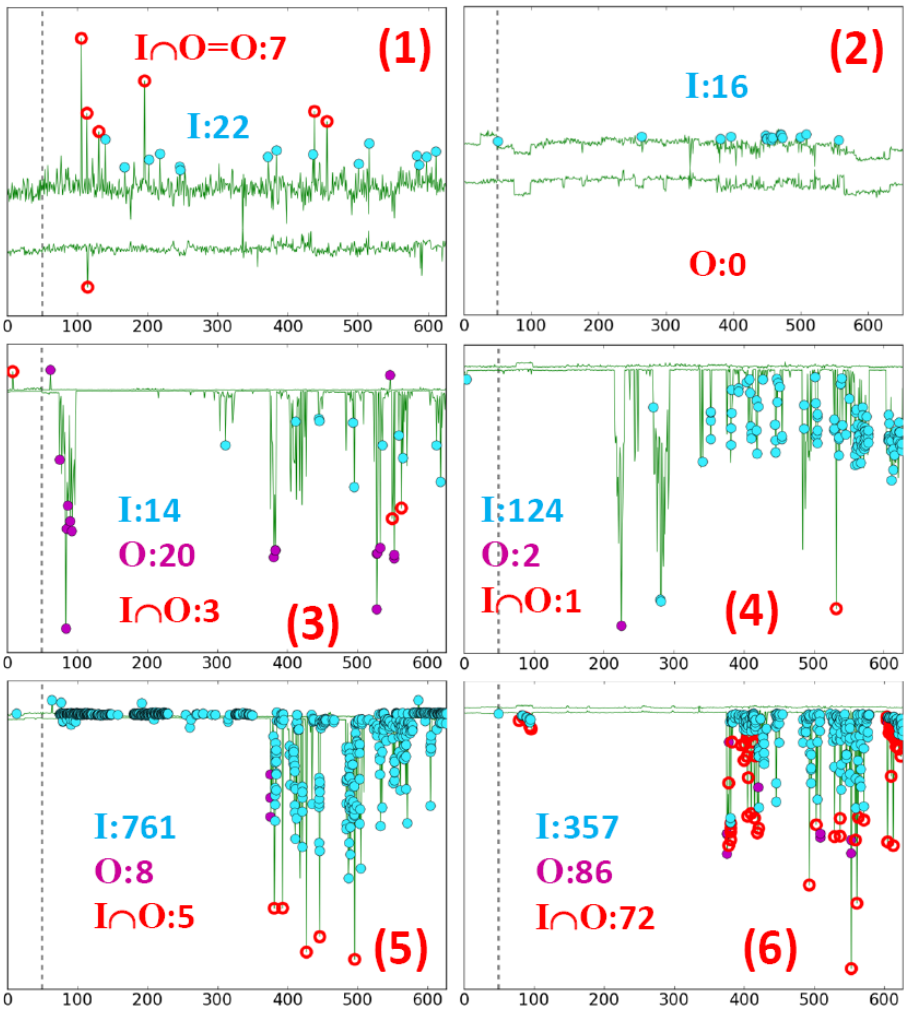


FIGURE 4.21: Interesting over-screen cases

Examples (3) and (4) show two cases where the online approach disagrees with the DPAT and AEC models. Nevertheless, the online outliers seem to make more intuitive sense than the outliers deemed by the DPAT and AEC models.

The 761 I-outliers in example (5) represent the largest number in Figure 4.20. Example (5) is similar to example (4) where online outliers make better sense while DPAT and AEC models over-screen many marginal outliers.

Example (6) shows the case where the number of online outliers is the largest. The number of I-outliers is also large. The **I**-coverage $= 100 * \frac{72}{86} = 83.72\%$. Notice that many

I-outliers are more marginal than the online outliers. However, this case also shows that even with the online approach, some marginal outliers still cannot be avoided.

Figure 4.22 shows the results of the remaining 450 tests in the under-screen category. The comparison focuses on the number of I-outliers (|**I**|), the number of online outliers (|**O**|), and the number of global outliers (|**G**|).

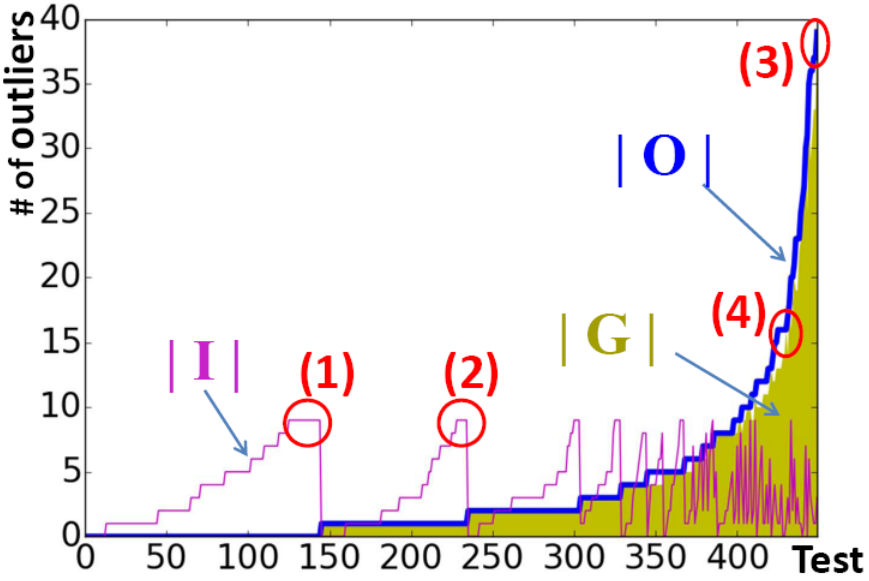


FIGURE 4.22: Result summary for under-screen cases

As illustrated in Figure 4.19 before, the online approach can find more outliers than a PAT model in the under-screen situation. Hence, for the under-screen category, the metric of interest is no longer the **I**-coverage. Instead, the focus is on the coverage on the global outliers, **G**-coverage $= 100 * \frac{|\mathbf{G}|}{|\mathbf{O}|}$. The average **G**-coverage across the 450 tests is 97.06% (excluding tests with zero online outliers), showing that most of the online outliers are also global outliers, a desirable property if global outliers are treated as the ideal answer.

Four examples are selected from Figure 4.22 for illustration. They are shown in Figure 4.23. All four examples have 100% **G**-coverage, so global outliers are not highlighted.

Examples (1) and (2) show two cases where $|\mathbf{I}| > |\mathbf{O}|$. Pictorially, the I-outliers in example (1) all appear to be marginal. The I-outliers in example (2) also appear to be marginal, except for the one shared outlier which is also the most outlying die.

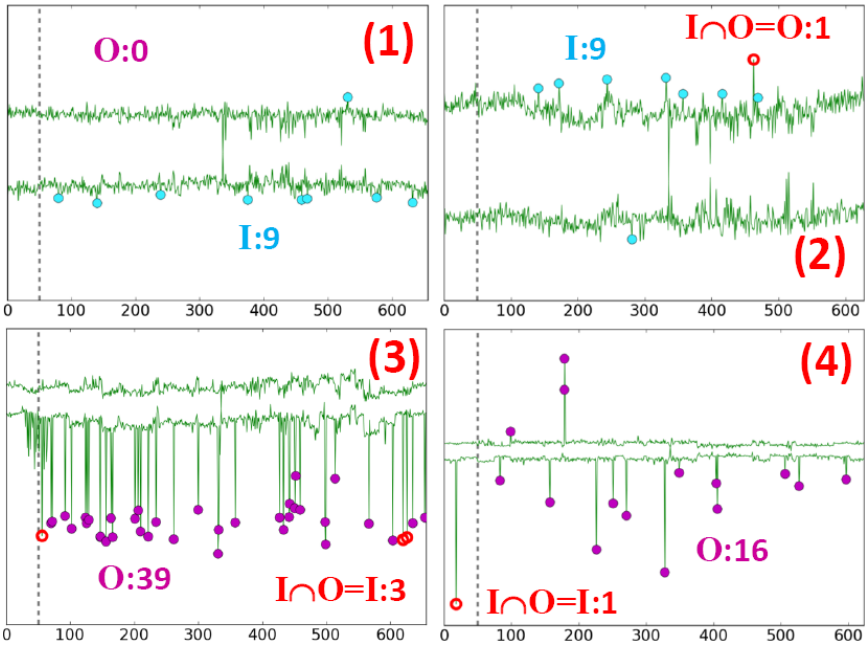


FIGURE 4.23: Interesting under-screen cases

Examples (3) and (4) then show two cases where $|\mathbf{I}| \ll |\mathbf{O}|$ (but the additional online outliers are reasonable). Example (3) is the case where the number of online outliers, 39, is the largest while there are only 3 I-outliers. Upon visual inspection, one can see that most of the online outliers are similar in their test values to the 3 shared outliers. There are 2 online outliers which are slightly more inlying. However, they are validated by being global outliers.

Example (4) depicts an instance where the DPAT and AEC models capture only one die (which is also the most outlying die). This die appears in an early wafer. The online approach captures 16 outliers, where pictorially the additional 15 online outliers appear to also deviate from the trend.

Results presented above are further summarized in Table 4.2. The table includes

three additional sets of results. The "MCU/Uni-v" shows results based on an automotive micro-controller with univariate analysis (Uni-v). The micro-controller data contains about 8K wafers. Hence, for finding N most similar wafers, let $N \leq$ 2K. Also let $b = 100$ wafers due to each wafer having fewer dies.

The notation "Multi-v" denotes the multivariate analysis where the **I**-outliers stand for the outliers shared by the Mah and LR methods. The probability-based online approach is applied with the LR method where instead of using a DPAT model on the "new measured values," the online approach is applied. Each model is a 2D model based on two correlated tests (> 0.9). No test is used in more than one model.

TABLE 4.2: Result summary on two automotive product lines

Product /*Case*	*Category*	#tests, or #models	|**O**| *ave*	|**O**| *max*	|**I**| *ave*	|**I**| *max*	Average G-Cov.	Average I-cov.
Sensor	*over*	306	8.65	86	45.91	761	60.57%	**90.95**%
/Uni-v	*under*	450	3.60	39	3.51	9	**97.06**%	57.67%
MCU	*over*	42	8.21	45	38.59	126	77.47%	**81.74**%
/Uni-v	*under*	218	8.09	49	5.61	19	**97.61**%	37.99%
Sensor	*over*	58	12.21	97	56.92	344	56.43%	**88.76**%
/Multi-v	*under*	108	5.69	44	3.41	9	**96.51**%	49.50%
MCU	*over*	5	12.6	58	39.0	83	60.49%	**99.14**%
/Multi-v	*under*	55	10.42	35	4.25	19	**92.44**%	13.21%

Results in Table 4.2 are also divided into over-screen and under-screen categories. For the over-screen category, observe that the average (ave) and the max (max) of the |**O**| numbers across tests are consistently smaller than the average and max of the |**I**| numbers. This shows that the online approach achieves reduced temporal uncertainty. The **I**-coverage (I-cov) numbers are between 80% and 99%, indicating small spatial uncertainty, though the I-cov results for the MCU/Uni-v are not as high as others, showing more divergence in the outliers found by the online approach and the DPAT and AEC models.

For the under-screen category, the **G**-coverage (G-cov) numbers are all above 92%,

showing most of the online outliers are global outliers. If global outliers are treated as the golden reference (see discussion in Sections 4.7.2 and 4.7.3 and examples (3) and (4) in Figure 4.23 before), then the online approach is an effective way to obtain those ideal answers.

As discussed earlier, the **I**-coverage for the under-screen category is not as meaningful. On the other hand, notice that the **G**-coverage numbers for the over-screen category are much lower than those for the under-screen category. This reveals that, as the online approach finds more outliers, more of them cannot be validated as global outliers. For online evaluation, this could be due to the lack of information contained in the future wafers.

4.9 Summary

In this chapter, the potential issues with the existing distribution-based outlier methods were explored. These issues were highlighted in terms of two concepts: temporal uncertainty and spatial uncertainty. It was then shown that these uncertainties can be reduced if an outlier approach is designed to find only the gross outliers. Additionally, a marginality test and a probability-based outlier measure are proposed to realize such an approach. Their benefits are demonstrated based on experiments using two automotive products. Note that in Section 4.8, online outliers were compared to **I**-outliers shared by DPAT and AEC models. If the comparison was instead based on the shared gross outliers (Section 4.4.5) from DPAT and AEC, the findings would be similar.

While the results from the proposed probability-based outlier method are very promising, in the scope of research contribution it is just another outlier method. The calculations required to implement it are quite expensive and it may be unlikely to be adopted into manufacturing flows. Therefore, though the method has theoretical and experimental backing, it may lack practical value. However, the discoveries about

some core aspects of outlier analysis made throughout this work inspired further research into the subject. The study in this chapter served as an important stepping stone toward developing the universal outlier model evaluation framework presented in Chapter 5.

Chapter 5

Consistency in Wafer Based Outlier Screening

5.1 Overview

Outlier screening is a popular approach for testing automotive products. In practice, developing an outlier model can be subjective, making justification of the model challenging. A new concept called *Consistency*, which provides a data-driven objective way to assess an outlier model, is proposed and described in this chapter. The development of outlier models in view of this consistency concept is studied and experimental findings are reported based on an automotive product line.

5.2 Introduction

Recall from Section 1.1.2 and Figure 1.2 that three components are considered in developing an outlier model. These components are the base set, the sample outlier scores, and the threshold.

The decision for classifying a sample as an outlier involves subjectivities. First, an outlier decision is made with respect to the samples in the base set. Altering this set

may change an outlier to an inlier and vice versa. In practice, when outlier screening is applied in wafer sort test, a common strategy is to let a base set include dies from the same wafer. It is then understood that the screening is looking for wafer-based outliers, in contrast to for example lot-based outliers. This is also the strategy considered in this work.

The choice of outlier score calculation can also be subjective. Many methods have been proposed for outlier score calculation. For example, Part Average Testing (PAT) is popularly used for automotive product lines [37]. PAT can include Static PAT (SPAT), Dynamic PAT (DPAT), Automotive Electronic Council DPAT (AEC), and Robust DPAT (RDPAT) [38]. PAT methods determine outliers based on measured values of the samples (dies) in the base set. They can be called *distribution-based* methods.

Different outlier score calculation can produce different outlier ranks. Two questions might be asked in practice: (1) Which outlier analysis method is the best? (2) If one method has been applied, what will be the next best method to apply? For example, the amount of variance reduction is a good metric to show merits of an outlier method [39][40][41]. If a univariate outlier screen has been applied, it is also shown that multivariate outlier screen can capture unique (and failure-analysis-verifiable) outliers [12][8].

There can be two basic strategies for determining the threshold. One strategy is based on known fails such as burn-in [41][10], wafer sort fails [13], and customer returns [8]. The other strategy is based on setting a yield reduction budget to be tolerated, e.g. 10 PPM [68]. If known fails are to be used, there is an issue of knowing whether or not the fails are sufficiently representative. If a yield budget is given, it can be challenging to foresee what level of yield reduction is optimal. The performance of an outlier model can change as characteristics in the test data change over time, as was demonstrated in Chapter 4, and hence, require model validation to adapt to the change [44].

The subjectivities in building an outlier model can make it challenging to justify the outliers. In practice, one way to justify the outliers is through detailed analysis verifying that some of the outliers are indeed defective parts. However, such analysis is both expensive and time consuming. A lagging indicator is great for confirmation, but deploying the screening method earlier is often desirable in order to protect the WIP and enable the continuity of supply. Moreover, there is also the question of whether defective parts are miss-classified as inliers. The practical strategy to approach this question typically depends on monitoring the customer return rate.

In this chapter, a new concept is proposed to provide another *statistical* perspective for justifying outliers. It is called *consistency*. Note that this consistency and the study presented alongside it are purely statistical, without referencing to actual defects. The consistency is based on the following assumption:

> Outlier decisions on one wafer should be *consistent* with outlier decisions on other *similar* wafers.

This assumption led to the development of a *consistency check* for outliers identified by a model. The check further classifies outliers into consistent outliers and inconsistent outliers. The goal of this work is therefore to understand what properties can be stated for those consistent outliers.

The rest of this chapter is organized as follows. In Section 5.3, three outlier methods are selected as examples, and are used to experimentally illustrate the inconsistency among their outlier decisions. Section 5.4 explains the proposed consistency check in detail. The main point to illustrate there is that ensuring consistency among outlier decisions across wafers using one method leads to improved consistency among outlier decisions across methods. Section 5.5 then focuses the discussion on determining the sets of *similar* wafers. A clustering based method is presented for detecting systematic

shifts in the wafer test data. The impact of applying this clustering method with the consistency check is discussed. Lastly, Section 5.6 summarizes the chapter.

5.3 Potential inconsistency among methods

As mentioned in the previous section, different outlier methods calculate outlier scores differently, resulting in different outlier ranks. Consequently, if a threshold is set to screen top j outliers, their outlier sets can be different.

To illustrate this inconsistency across methods, three outlier methods are selected as examples. The first method is a variation of the distribution-based method SPAT. The second is the distribution-based method DPAT. The third is a variation of the hybrid distribution and location-based method called location averaging (LA). It is important to note that the purpose of this study is not trying to assess which method is better. Rather, the methods are simply used as examples to illustrate the inconsistency.

Suppose a wafer contains n dies whose measured values from a single test are $\{m_1, \ldots, m_n\}$. Let their mean value be denoted as μ. In the particular SPAT calculation considered, the outlier scores are simply $\{s_1 = |m_1 - \mu|, \ldots, s_n = |m_n - \mu|\}$. A threshold T_s is set such that if $s_i > T_s$, m_i is classified as an outlier.

Let the standard deviation of the measured values be denoted as σ. In the DPAT calculation, the outlier scores are $\{d_1 = \frac{s_1}{\sigma}, \ldots, d_n = \frac{s_n}{\sigma}\}$. A threshold T_d is set such that if $d_i > T_d$, m_i is classified as an outlier.

For calculating LA-like outlier scores, a window size is chosen. For example, a 9×9 window would include 81 dies. Then, another parameter k is set. For a given die with value m_i, first the 80 dies surrounding the die are identified. The number of dies may be smaller if the die resides near a wafer or if there are missing values. Among them, the k dies with closest values to m_i are identified. For example, $k = 40$. The mean of

these 40 values is calculated as μ_i. The outlier score is then calculated as the *residual* $l_i = |m_i - \mu_i|$. A threshold T_l is set such that if $l_i > T_l$, m_i is classified as an outlier.

The SPAT and DPAT methods are described based on the assumption that the distribution of $\{m_1, \ldots, m_n\}$ is somewhat symmetric. If it is asymmetric, adjustment can be made to the outlier scores by normalizing the scores on the left of the distribution with distance of the x quantile point p_x to the mean of the distribution, and on the right of the distribution with the distance of the y quantile point p_y to the mean of the distribution. For example, one can use $x = 1\%$ and $y = 99\%$, which are similar to those used in an AEC DPAT implementation [38]. On the other hand, LA scores are inherently immune to the asymmetric issue because the LA residual is a rank statistics based on non-parametric modeling of a distribution [40][69].

5.3.1 Two test examples

The wafer test data used for the study in this chapter came from a recent automotive SoC product line. The data comprises roughly 5000 wafers and 3 million parts. Because the focus of the study is to understand the consistency issue, it does not consider application of an outlier model in an online fashion as that studied in Chapter 4. Instead, all wafers are used in the study collectively. In other words, this work does not consider *temporal uncertainty* [68], i.e. the discrepancy between the performance of an outlier model on initial qualification lots versus the performance of the model on future lots.

To illustrate inconsistency among methods, the experiment is conducted as follows. An outlier method calculates an outlier score for each of the 3M dies. Thresholds are chosen to screen out the top j dies, for $j = 300$, 30, and 3, which correspond to 100 PPM, 10 PPM, and 1 PPM yield reduction, respectively. Figure 5.1 and Figure 5.2 show the results based on two test examples.

Observe in Figure 5.1-(c) that for $j = 3$, the three methods agree on which three dies should be screened. However, when j increases to 30 in (b), they agree on 20 out

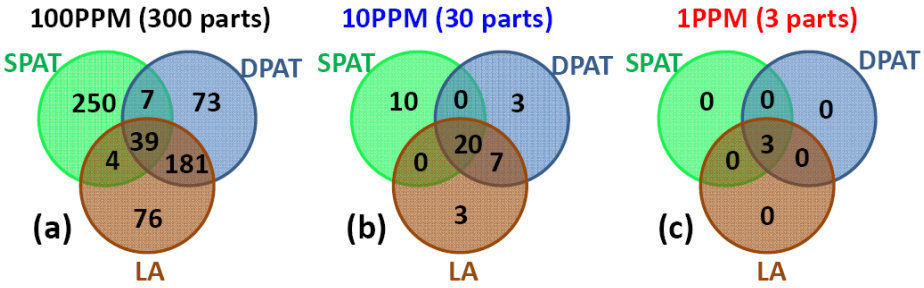


FIGURE 5.1: Test example 1

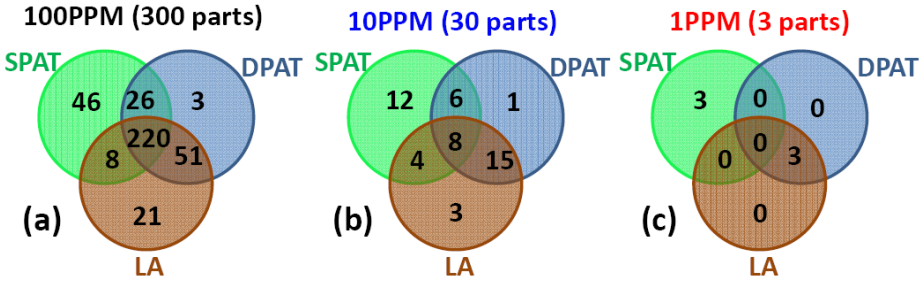


FIGURE 5.2: Test example 2

of the top 30 outliers, about 66%. When $j = 300$ in (a), the agreement drops to 39, or 13%.

Figure 5.2 shows another example where the trend reverses. When $j = 3$, no outlier is shared by all the three methods. When $j = 30$, 8 outliers are shared by the three methods, about 27%. When $j = 300$, 73.3%, or 220 outliers are shared. In both examples, the three methods do not always agree on which top j outliers to screen, depending on the setting of j.

To illustrate what test data characteristics might contribute to the opposite trends, Figure 5.3 shows two plots for test example 1, each plotting the minimum and maximum test values (green) on each wafer across all chronologically ordered wafers. The two plots differ on their vertical scale.

The first plot shows that there are 7 dies whose test values are well above the rest (circled red). Those 7 dies are so outlying that they are easy to be identified as outliers. In Figure 5.1-(c), when the three methods were asked to find the top 3 outliers, all of

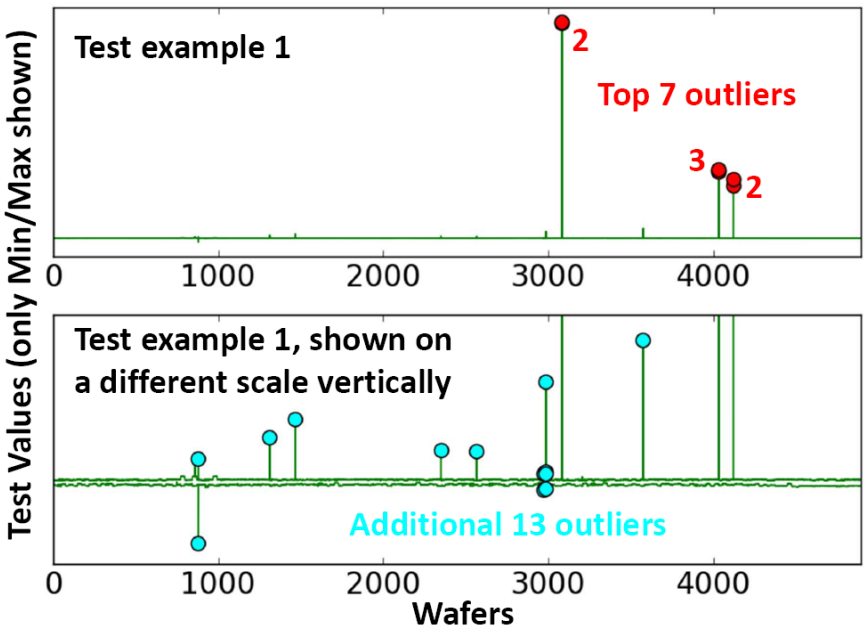


FIGURE 5.3: Min/Max value plot for test example 1

them found the highest two dies. The values of the next three top dies are close to each other. While they are visually indifferentiable in the plot with the particular vertical scale, the three methods identify the same die as the 3rd outlier, resulting in Figure 5.1-(c).

The second plot is with a different vertical scale so that next batch of outlying dies can be visualized more easily. Figure 5.1-(b) shows that there are 20 shared outliers. Excluding the top 7 outliers, the second plot shows where the remaining 13 shared outliers reside. Most of them are visually easy to accept as outliers while a few are not.

Figure 5.3 reveals that if a die is "clearly" outlying, it should be easy to identify and hence, it can be found by all methods, resulting in the sharing. The converse is not true. A shared outlier can still be "marginal" - The sharing can happen, depending on the statistics in the data and how different methods utilize the statistics.

For test example 2, Figure 5.4 shows a similar plot. Unlike test example 1, in this plot, there are no clearly outlying dies. This characteristic difference hints a reason for

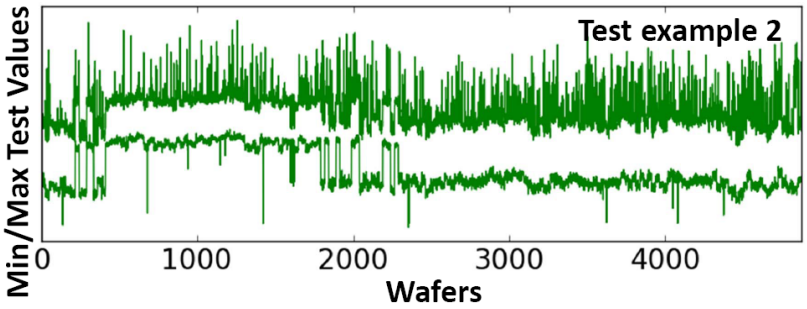


FIGURE 5.4: Min/Max value plot for test example 2

the opposite trends observed in Figure 5.1 and Figure 5.2.

Conceptually, if two dies on two different wafers have test values m_1 and m_2 where $m_1 \approx m_2$, one method can calculate the two outlier scores such that m_1 is treated as more outlying than m_2 while another method may decide that m_2 is more outlying than m_1. This is because different score calculations utilize different statistics in the data. For example, the SPAT calculation utilizes the mean. DPAT utilizes the mean and standard deviation. LA utilizes the mean of the closest test values in a spatial window.

Therefore, if the difference between m_1 and m_2 is not large enough to offset the difference in the score calculations, their ordering in the outlier rank by one method can be reversed by another method, leading to different outlier classification by the two methods.

The discussion above indicates that "very gross" outliers tend to be shared by different methods. Beyond those, it is difficult to say one way or the other. All that is known is that one method can disagree with another on their outlier sets, even for a small PPM reduction level.

5.4 Consistency check

Suppose an outlier screening methodology is constrained to use only one method. Could we know what is an optimal threshold, e.g. an optimal j value, such that the

top j outliers would likely be agreed by other outlier methods (without running those methods)? This motivates the consistency check discussed below.

Figure 5.5 illustrates the setting for consistency check. Suppose there are N dies across W wafers $w_1, \ldots, w_W$. Given a method, a die with test value m_i is converted into an outlier score s_i based on some statistics on the wafer the die is located on. The outlier scores across all dies are comparable and hence, a global rank can be obtained. A threshold is then set on this global rank to identify outliers. This is how, conceptually, outlier methods operate and the experiments in Section 5.3 were performed.

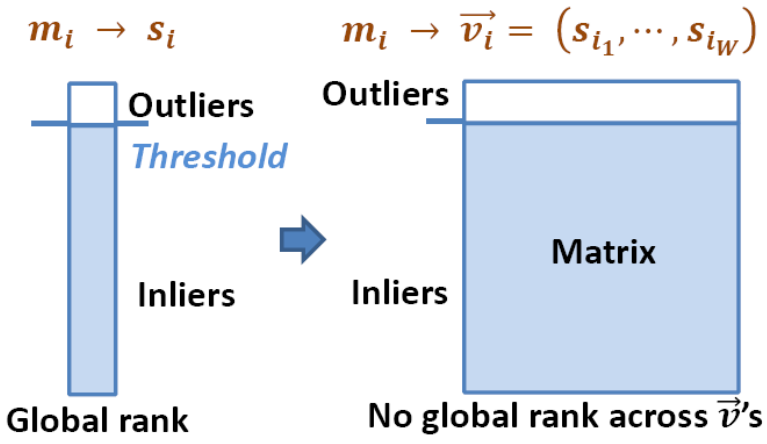


FIGURE 5.5: Illustration of the setting for consistency

In the new setting, each test value m_i is converted into a vector $\vec{v_i}$ of W scores $(s_{i_1}, \ldots, s_{i_W})$. Each s_{i_h} is calculated based on the statistics on wafer w_h. In other words, the die m_i is assumed to be on the wafer w_h for calculating s_{i_h}.

Given two outlier score vectors $\vec{v} = (v_1, \ldots, v_W)$ and $\vec{u} = (u_1, \ldots, u_W)$, $\vec{v} > \vec{u}$ is only true if $\forall i, v_i > u_i$, (or $\geq$). Then, given a threshold, suppose a die with $\vec{v_o}$ is classified as an outlier and a die with $\vec{v_i}$ is classified as an inlier. This classification is said to be *consistent* if $\vec{v_o} > \vec{v_i}$.

A given threshold is said to be *consistent* if $\vec{v_o} > \vec{v_i}$ holds true $\forall \vec{v_o}$ and $\forall \vec{v_i}$. In this case, the outliers themselves are said to be consistent outliers, and the outlier decision

is consistent.

Figure 5.5 illustrates that given N vectors, $\vec{v_1}, \ldots, \vec{v_N}$, for N dies, there no longer exists a single global rank across all dies. On this matrix, the objective is therefore to search for a *minimum* threshold such that the threshold is consistent.

5.4.1 Finding minimum consistent threshold

Earlier it was shown, based on two test examples, that different methods can disagree on their outlier sets for a given yield reduction level. Figure 5.6 includes more results to illustrate the same point. The figure contains 250 tests. For each test, the percentage of shared outliers among the three methods is shown for two cases, the top 1 PPM (3 dies) and the top 100 PPM (300 dies). For example, for 1 PPM, if there is 1 outlier shared, then the percentage shown is 33.3%.

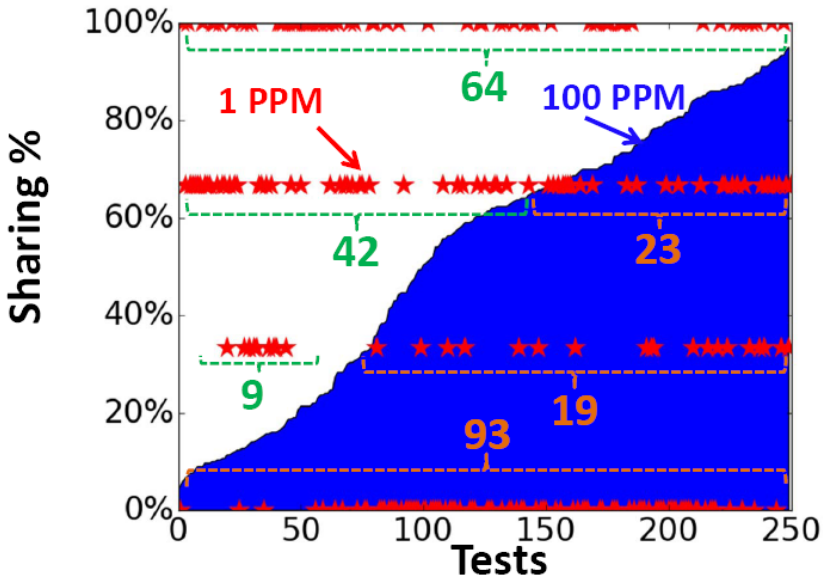


FIGURE 5.6: % of shared outliers by all three methods

Along the x-axis, the tests are ordered by their percentages from the 100 PPM case. The smallest percentage is 4.3% and the largest is 95%. The results are shown as a blue

region. For 1 PPM, there are four possible percentages, 0%, 33.3%, 66.6%, and 100%. The results are shown as red stars.

For 100 PPM, the inconsistency exists in every test, i.e. none has 100% sharing. The percentages for the 1 PPM case can be divided into two categories: those above the blue curve representing a trend similar to Figure 5.1 and those below the blue curve representing the reverse trend similar to Figure 5.2. The two cases contain 115 tests above the curve and 135 tests below.

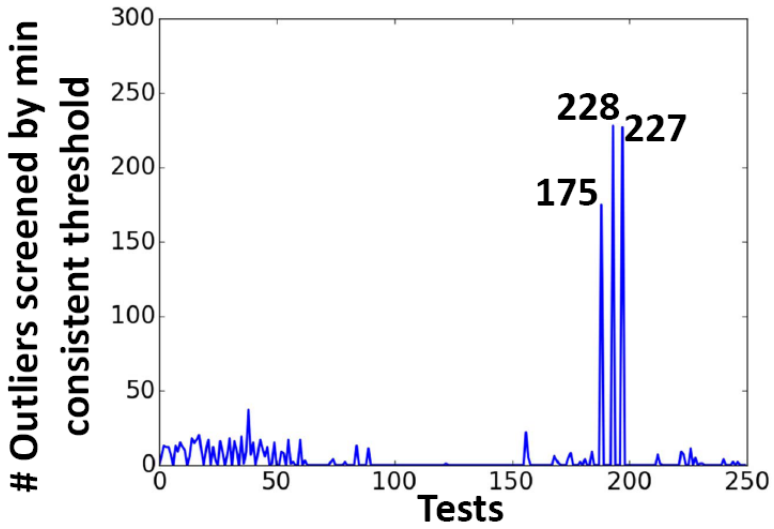


FIGURE 5.7: Results with minimum consistent threshold - DPAT

Next, Figure 5.7 shows the number of outliers identified by the *minimum consistent threshold* for each test based on the DPAT method. The tests are presented in the same order as in Figure 5.6. This number is always smaller than the number of shared outliers for the 100 PPM case shown in Figure 5.6.

Suppose that for a test t, the number of consistent outliers shown in Figure 5.7 is c. Then, for every test, these c consistent outliers from DPAT are exactly the same top c outliers classified by SPAT and LA. In other words, every outlier of the c consistent outliers from DPAT is agreed by the SPAT and LA methods as one of their respective top c outliers.

LA

If LA was used instead of DPAT in the experiment associated with Figure 5.7, the sets of outliers would be exactly the same as DPAT results for 243 out of the 250 tests. The numbers of consistent outliers found by each method for the remaining 7 tests is reported in table 5.1 below.

TABLE 5.1: DPAT and LA disagree on consistent outliers

Test index	149	151	174	175	185	218	220
# DPAT-consistent outliers	7	0	3	0	0	11	10
# LA-consistent outliers	9	10	5	1	8	12	24

Recall that every DPAT-consistent outlier is LA-consistent outlier. Hence, for those 7 tests, LA found additional consistent outliers.

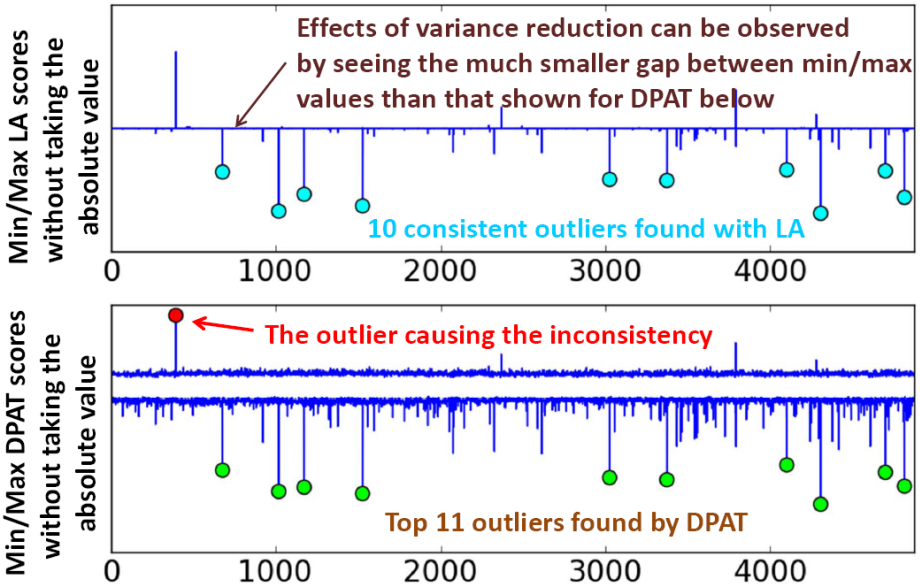


FIGURE 5.8: Illustration of test index 151 result

Take test index 151 as an example. Figure 5.8 first shows where the 10 LA-consistent outliers locate in the LA space where min/max LA scores are shown for each wafer.

Then, the second plot is shown in the DPAT space, where the top 11 DPAT outliers are inconsistent because of one outlier highlighted in red. If this outlier was removed, the remaining 10 would become DPAT-consistent by themselves. The underlying reason behind this is that DPAT treats outliers on the two sides of the distribution together while LA treats them separately, resulting in LA finding additional consistent outliers.

The seven tests show that there can be aspects of difference between two methods which are not resolved by the consistency check. Nevertheless, for most tests, consistency check ensures that outliers found by DPAT and LA are mutually consistent.

SPAT

Figure 5.9 shows the number of SPAT-consistent outliers. SPAT agrees with DPAT on consistent outliers for 121 tests. For the remaining 129 tests, there are more SPAT-consistent outliers.

Let C_d, C_l, C_s be the sets of consistent outliers from DPAT, LA, and SPAT on a test, respectively. It is important to note that in the results for the 250 tests under consideration, $C_d \subseteq C_l \subseteq C_s$ is always true.

In Figure 5.9, there are 33 tests with more than 400 SPAT-consistent outliers. For 32 of them, DPAT has zero consistent outliers. These 32 tests happen to be the same class of opens tests with similar behavior.

Noise band

Figure 5.10 uses one of the 32 tests to explain what happens. For a test value m_i, recall from Figure 5.5 that consistency check calculates a vector $\vec{v_i} = (s_{i_1}, \ldots, s_{i_W})$. Without loss of generality, assume s_{i_1} is the original outlier score s_i. Let $s_{min} = \min\{s_{i_2}, \ldots, s_{i_W}\}$. A *noise band* for the die is defined as: $\mathcal{N}_i = s_i - s_{min}$.

For a given threshold T, when m_i is classified as an outlier, it means that $s_i > T$. In consistency check, if $(s_i - \mathcal{N}_i) \leq T$ then s_i becomes inconsistent. In other words, for a

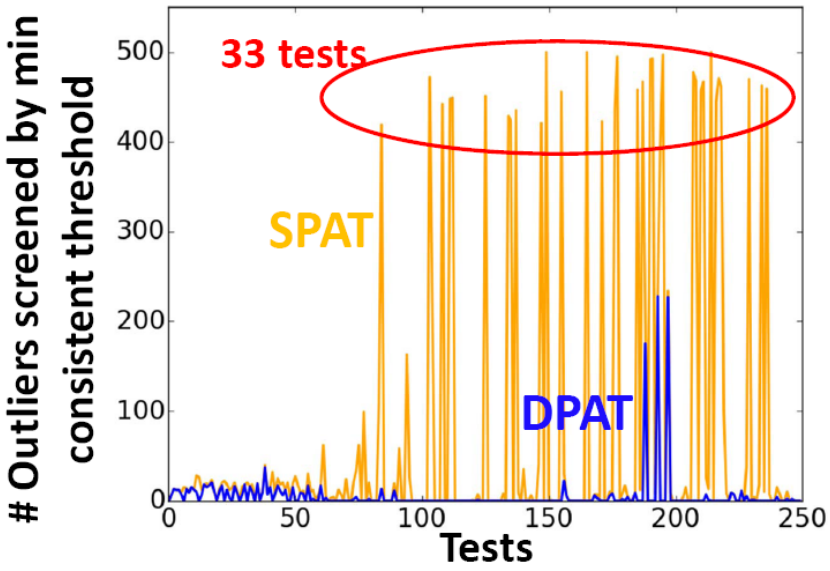


FIGURE 5.9: # of SPAT-consistent outliers (vs DPAT)

threshold to be consistent, every outlier must satisfy the property that its outlier score minus its noise band is still greater than T.

Figure 5.10 plots one hypothetical threshold in the SPAT space and another in the DPAT space, with noise bands shown only for the hypothetical outliers. Notice that SPAT noise bands are much smaller and hardly observable in the plot. In contrast, DPAT noise bands are much larger. Consequently, it is much easier for SPAT to find a vertical "gap" in the space where none of the noise bands would touch the threshold. On the other hand, DPAT could not find a threshold without any noise band touching it, leading to zero consistent outliers.

The source of a noise band is the *wafer-to-wafer* variation in the statistics used to calculate the outlier scores. In SPAT, this statistic is the *mean* of the distribution. In DPAT, it is the *mean* divided by the *standard deviation*. Hence, the DPAT calculation is subject to more noise.

The fact we have $C_d \subseteq C_l \subseteq C_s$ indicates that DPAT calculation has more noise than LA calculation which in turn has more noise than SPAT. This result suggests that if the

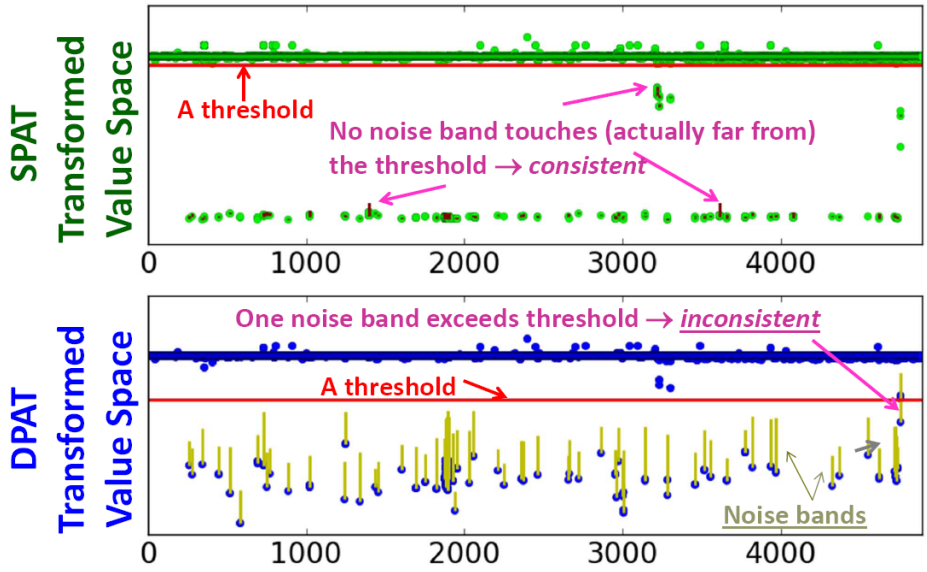


FIGURE 5.10: SPAT noise bands vs. DPAT noise bands

intention was to choose a method such that the consistent outliers from the method are also to be consistent from the perspectives of other methods, one would choose the method with the largest noise.

5.5 Detecting systematic shift

In Figure 5.7, DPAT finds zero consistent outliers on 173 tests. Because consistency check depends on wafer-to-wafer statistical variations, it is possible that capturing zero consistent outliers is due to systematic shifts in the statistics. To address this issue, a clustering based method is developed for detecting systematic shifts.

Given a set of wafers, the first step is to project them into a space defined by some selected features of statistics, for example mean and standard deviation. Clustering is then applied in this space.

It is well-known that for unsupervised learning such as clustering, model selection is one of the biggest issues. For example, with the K-means algorithm, determining the

k is a key consideration. The model selection in this work adopts the idea of *empirical risk minimization* (ERM) [70].

Given a set of wafers, begin by dividing them into two sets S_1, S_2. A clustering model M_1 is built for S_1 and a clustering model M_2 is built for S_2 by applying the same clustering algorithm. A mapping is used to map every wafer in S_2 to a wafer in S_1. For example, this mapping is based on the nearest neighbor. With this mapping, wafers in S_2 are re-clustered using the model M_1. The result is a new model M_2'. A *model stability* is calculated as the percentage of wafers that M_2 and M_2' agree on.

To avoid statistical bias in stability calculation, a Monte Carlo process is involved to randomly sample S_1, S_2 many times and calculate an *average model stability* (AMS). The AMS is then used to select the model. To obtain a clean partitioning on wafers, the desired resulting model will have a very high AMS, e.g. $>$99.9%.

Computing the AMS can be computationally expensive. Attention is needed to optimize the process. For example, the KD-tree was used to find the nearest neighbors [71] which obtained orders-of-magnitude speedup. Since the focus here is to assess the impact of systematic shifts to consistency check, details of the clustering implementation are omitted.

Figure 5.11 shows two examples. On the left two plots, every dot represents a wafer. In the first example, two clusters are found, colored red and green. The AMS measure is 99.93% for this clustering model. When the two clusters are seen in the wafer-to-wafer temporal view on the right (shown with the same cluster colors) where each wafer's $[\mu - \sigma, \mu + \sigma]$ range is shown in blue, systematic shifts can clearly be observed, matching the clustering result.

The second example has five clusters. The AMS measure is 99.97% for this model. The temporal view shows that four clusters are from the early wafers. After about 550 wafers, all remaining wafers belong to the same cluster (yellow).

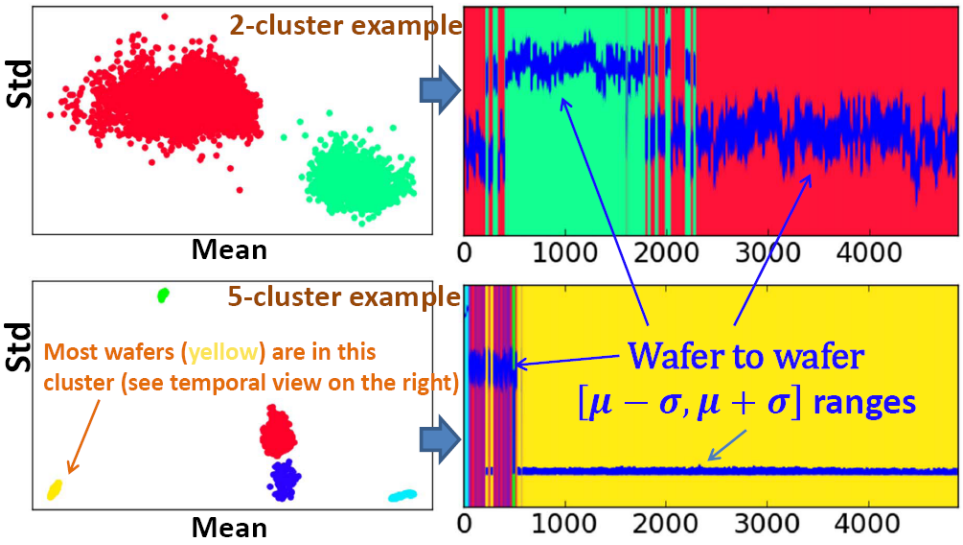


FIGURE 5.11: Two examples to illustrate clustering results

5.5.1 Impact of clustering on consistency check

After applying the clustering, DPAT finds more consistent outliers for 68 tests while finding the same number of consistent outliers for the other 182 tests. Out of these 68 test, 54 are among the 173 tests where DPAT found zero consistent outliers before (Figure 5.7). Hence, even after the clustering, 119 tests still have zero DPAT-consistent outlier.

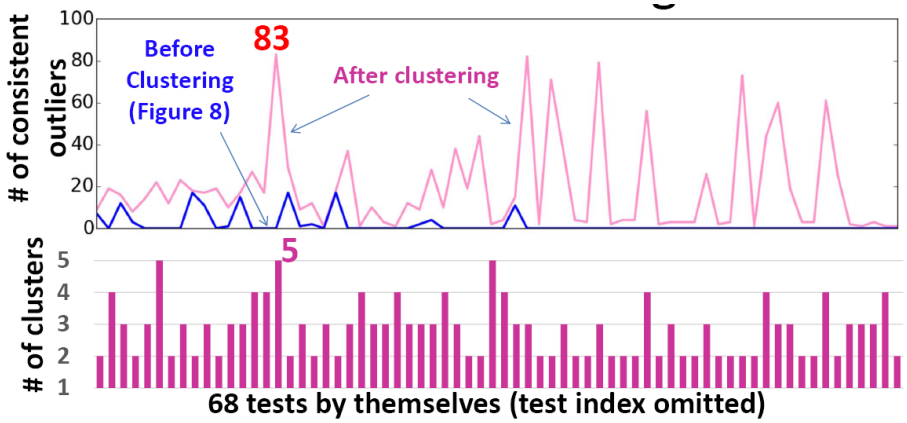


FIGURE 5.12: Impact of clustering on consistency check

Figure 5.12 shows the results on the 68 tests. The top portion shows the numbers of outliers before clustering in blue and the numbers of outliers after clustering in pink. The bar graph in the bottom portion shows the resulting number of clusters for each test. The test shown with 83 consistent outliers is the 5-cluster example shown in Figure 5.11 earlier. This is also the test with the largest number of consistent outliers among the 68 tests. This test had zero DPAT-consistent outliers before.

5.5.2 Finding no DPAT-consistent outlier

Figure 5.13 shows an example from the 119 tests with no DPAT-consistent outliers after clustering. The left plot shows one cluster and the right plot shows there is no clear shift of the mean and standard deviation in the temporal view. In other words, the clustering method did not detect a systematic shift in the data and this finding can be visually verified. In such a case, the result with clustering is identical to the result without clustering.

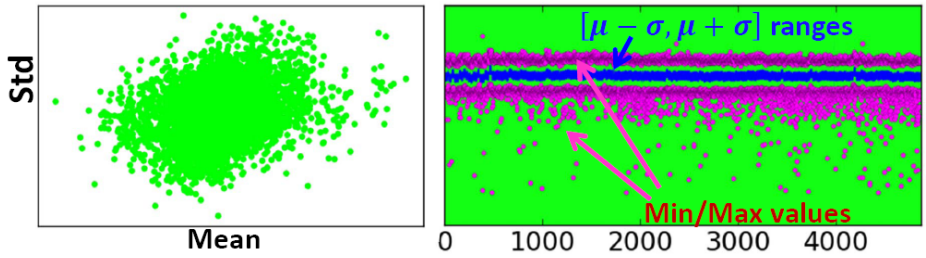


FIGURE 5.13: An example of no DPAT-consistent outlier

The right plot also shows the min/max values as purple points. From the discussion in Section 5.4.1, finding a consistent threshold is akin to finding a large enough vertical gap in such a plot, where the length of the gap is at least as long as the shortest distance between an inlier and an outlier. On the right plot, it can be observed that the dies are densely distributed and it is hard to obtain such a vertical gap. This visualization may help to intuitively understand why DPAT finds no consistent outlier.

It is important to note that the minimum consistent threshold can be used as a lower bound on the threshold. This lower bound enables the assertion that some tests have no outliers with a particular method. Such a property can be very desirable in practice, allowing the analysis to identify situations where there is no outlier.

5.6 Summary

In this chapter, a new perspective called consistency was proposed for justifying outliers. The consistency check enables an outlier method to find a minimum threshold using a large number of wafers to obtain consistent outliers. When a method with large noise in the outlier score calculation is used with the consistency check, the resulting consistent outliers are likely to be agreeable by other methods as their respective consistent outliers.

A clustering method was proposed to detect systematic shifts which can alter the results of the consistency check. While in the experiments the shifts are with respect to the mean and standard deviation, other features, such as wafer spatial patterns, can be used to detect different types of shifts.

This study was carried out by analyzing all wafers together (including the clustering experiments). In practice, wafers come one by one. Hence, how to implement the consistency check and the clustering based systematic shift detection in an online fashion is the next interesting question.

While the objective of this work is not to compare different methods, the results seem to provide another new perspective to say that LA is better, as suggested in earlier works [41][69]. On one hand, LA finds much fewer consistent outliers than SPAT. On the other hand, LA finds a few additional outliers over DPAT. In this sense, consistency could serve a purpose as a theoretical tool to compare different methods.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The work in this thesis explored the effect of subjectivity on data analytics in Test. In Chapter 1, it was shown that many of the applications of data analytics in Test can be reduced to correlation analysis or outlier analysis. Chapters 2 and 3 focused on correlation analysis, while Chapters 4 and 5 focused on outlier analysis.

The added value of data analytics in Test was clearly demonstrated using a real production yield issue in Chapter 2. A novel methodology using advanced statistical correlation methods was applied to optimize production yield for an automotive product line, and the result was confirmed by a silicon split-lot experiment. The yield optimization was successful despite previous failed efforts carried out by the test, design, and yield analysis teams. This finding supports the claim that effectiveness of yield optimization efforts through data analytics is dependent on the experience of the analyst.

An important conclusion to draw is that the data analytics process is not automatic even if efficient state-of-the-art analytic tools are available. The analytics steps where subjectivity was recognized to originate in correlation analysis were data preparation and meaningfulness determination (in the process in Figure 3.1). Chapter 3 proposed

an approach for learning the process of correlation analysis by using a process mining (PM) model. The effectiveness of the approach was demonstrated by applying the PM model learned from resolving the yield issue in one product line to a yield issue in another automotive product line.

Chapter 4 explored the potential issues with distribution-based outlier methods and introduced the concepts of temporal uncertainty and spatial uncertainty as measures of those issues. A marginality test was proposed to differentiate between marginal outliers and gross outliers, and it was shown that focusing only on gross outliers lead to a reduction in both uncertainties. Additionally, a new probability-based outlier method was proposed and the findings were demonstrated using data from two automotive products.

Lastly, Chapter 5 proposed a methodology called consistency, which uses wafer-to-wafer noise to justify outliers. The methodology is equipped with a check that can evaluate outlier thresholds across a large number of wafers to obtain consistent outliers. The idea behind consistent outliers was that true outliers should still be seen as outliers when projected onto other wafers. Experiments corroborating the methodology were conducted on an automotive SoC product. One interesting finding was that even though identifying outlier sets that were agreed upon by different outlier methods was not a direct goal of the methodology, a containment property across methods held true for the consistent outliers in the experiments. This finding suggests that consistency inadvertently reduces spatial uncertainty, which is a desirable property for an outlier analysis methodology.

6.1.1 Subjectivity Reduction in Correlation Analysis

Integration of the PM based approach proposed in Chapter 3 into a semiconductor production process would directly target and reduce the subjectivity in yield optimization.

A large portion of the subjectivity in correlation analysis stems from the data preparation step. The proposed approach could be used to develop a tool that is able to learn and reproduce the data preparation steps conducted by multiple analysts. A less experienced analyst could then apply correlation analysis as if he or she had the previous analysis experience that was learned by the tool.

The work of utilizing PM for correlation analysis is a first step towards autonomous analytics. The main idea is to learn from past usage experience and apply what is learned to future analytics problems. The biggest challenge with enabling this type of approach is the careful design and definition of the process steps. A set of process steps must be designed to be modular and expandable. If the set was immutable, subjectivity would exist with respect to the set itself. The work in Chapter 3 focused specifically on yield optimization where, although already difficult, designing such a set was relatively simple. This work serves as a proof of concept that learning from experience in correlation analysis is viable and is effective at reducing subjectivity.

An important property of the proposed approach is that it does not invalidate any prior work done in this field. On the contrary, a tool based on this approach would enable for easier integration of novel solutions into existing yield optimization flows. It makes sense for an analyst to prefer using methods and data preparation steps that led to successes in previous analyses. When a novel solution is proposed by another analyst, it may lack some of the steps that contributed to prior analyses being successful. A tool implementing the proposed approach would only need to be updated with the process steps relevant to the novel solution. Then, the tool could produce a generalized analytics path containing the various known good steps combined with the new process steps.

6.1.2 Subjectivity Reduction in Outlier Analysis

Recall that the main sources of subjectivity in outlier analysis were identified to be due to the choice of the base set, choice of the outlier scoring method, and threshold selection. The proposed consistency check based methodology targets the subjectivity due to threshold selection. By utilizing a large number of wafers, consistency can objectively search for thresholds that separate the true outliers from inliers.

The experiments in Chapter 5 showed that when a method with large noise in the outlier score calculation is used with the consistency check, the resulting consistent outliers are likely to be agreeable by other methods as their respective consistent outliers. In that sense, the subjectivity from outlier method selection can also be reduced through applying consistency. Being aware of what type of noise each method is prone to be affected by may allow an analyst to select methods that are more likely to consistently screen outliers at the desired yield reduction budget.

Consistency effectively removes most of the subjectivity due to threshold selection and some of the subjectivity due to the choice of outlier method. However, the subjectivity due to base set selection remain unaffected. Although some subjectivity still remains, the present reduction in subjectivity is already beneficial, as shown through real data examples in Chapter 5. This result confirms that subjectivity reduction is the correct objective and that further work in this direction is likely to further benefit outlier analysis.

Because consistency is designed to be applicable with any method, prior and future works introducing novel outlier methods are still valuable. The availability of consistency henceforth allows future development to focus on aspects other than the subjectivity in threshold selection. As such, it may be easier for future methods to further improve the robustness of outlier analysis by explicitly considering the subjectivity of base set selection.

One very significant way in which consistency enables robust outlier analysis is by making it possible for any outlier method to reach the conclusion that no outliers exist in the data. This distinction is a key contribution because consistency can be integrated into any outlier method, thereby removing the subjectivity of threshold selection from any method. Generally for most methods, outlier scores have a direct mapping to the outlier rankings, leaving the decision for where to draw the cutoff to the analyst. Consistency entirely eliminates that decision by using a global data perspective to evaluate the threshold.

6.2 Future Research Directions

The work described in this dissertation explored the sources of subjectivity in correlation analysis and outlier analysis. The disadvantages of the subjectivities were demonstrated and solutions were proposed to reduce some of those subjectivities. Since the need for robust data analytics was shown and many problems still remain unsolved, a number of promising future research directions exist that may lead to significant advances in Test.

The work in Chapter 3 showed how a PM algorithm can be used to learn the process of correlation analysis for yield optimization. As mentioned before in Section 3.8, one limitation of the current PM approach is that cross-step dependency was not considered. Special consideration was taken in designing the process steps to avoid such a dependency. In future work, the PM algorithm may need to be enhanced to explicitly take cross-step dependency into account. Another drawback of the proposed PM approach is that it does not allow loops in the process. It is conceivable that some analytics processes will contain loops, and accounting for those will substantially increase the complexity of the algorithm.

Additionally, the applicability of a PM approach to other types of data analytics applications is not guaranteed. The solution with respect to yield optimization was mainly focused on the dataset construction step in the analytics search process pictured in Figure 3.1. The meaningfulness determination step may require further research to become compatible with a PM approach.

The biggest contribution to outlier analysis presented in this dissertation is the consistency concept from Chapter 5. However, that study was done by analyzing all wafers together, where a true global view of the data was available. In practice, future wafers would not be known. Modification to the consistency check to enable such a practical scenario is an important future work. Another modification that can be made to consistency would be to change the clustering method used for removal of systematic variation. Other features besides the mean and standard deviation can be used to detect clustering, and the clustering would also be affected by the modification made for the consistency check to be applicable in an online fashion.

An important consideration for any outlier model would be uncovering the physical meaning behind a defect model. With the consistency check, it is possible for a defective part to be categorized as an inconsistent outlier. The consistency check is not proposed to ensure capturing all defects. Rather, it is a way to differentiate statistically easy-to-justify ones (consistent outliers) from hard-to-justify ones (inconsistent outliers). Correlating them in terms of a defect model is an important future work.

6.2.1 Learning the Process of Outlier Analysis

One obvious future work would be to set out to learn the process of outlier analysis. Combining the insights gathered from developing a PM approach for yield optimization and the consistency check sounds like the logical next step. However, yield optimization was selected as the flagship analytics application for PM for a reason. The

scope of the yield optimization problem is less complex and more controllable than the scope of many outlier analysis applications.

Though the result for yield optimization is encouraging, generalizing the approach to an outlier analysis application such as customer return analysis will require enhancement of the current PM design. As mentioned before in Section 3.8, there is a trade-off between the objective to simplify the PM algorithm and the objective to allow flexibility in designing the process steps. The capability of a PM model is limited by the set of process steps. Devising the modular process steps required for a functioning PM implementation will require careful planning.

The proposed consistency check would play a key role in learning the process of outlier analysis. As it was stated earlier, the main steps of the analytics process to be learned are dataset preparation and meaningfulness determination. In a sense, consistency could be directly used as the meaningfulness determination component of outlier analysis. This approach would simplify the implementation of the PM algorithm. Generalization of the PM approach to outlier analysis is an interesting future work that will serve as another stepping stone towards autonomous analytics.

6.2.2 Applicability of Outlier Methods

Though consistency is an effective concept for assessing outlier models, it may not be sufficient on its own. Recall that the high-level idea behind consistency was to make outlier analysis immune to noise introduced from wafer-to-wafer variations. However, the information contained in wafer-to-wafer noise may not contain information about the shape of the distribution on each wafer. In other words, even if all wafers have similar distributions, that distribution may violate some underlying assumptions of outlier methods.

For example, the DPAT method is intended to be applied on normal distributions.

For applying DPAT, in addition to checking consistency across wafers, it may be desirable to check the normality of the distributions. Such a check would be used to ensure that application of the outlier method is *justifiable*.

The concepts of consistency and justifiability could be used together to define a notion of *applicability* for outlier models. Some initial work into this notion has already been done, and any interested reader is advised to please refer to a paper titled "Some Considerations on Choosing An Outlier Method for Automotive Product Lines" that was submitted to the International Test Conference 2017. The work in that submission is an example of how future works can expand on the work in this dissertation in pursuit of robust data analytics.

Bibliography

[1] L.-C. Wang, "Experience of Data Analytics in EDA and Test - Principles, Promises, and Challenges", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, no. 99, 2016.

[2] W. B. Nelson, *Accelerated Testing: Statistical Models, Test Plans, and Data Analysis*, 1 edition. Wiley-Interscience, 2004.

[3] K. M. Butler, A. Nahar, and W. R. Daasch, "What We Know After Twelve Years Developing and Deploying Test Data Analytics Solutions", *ITC*, 2016.

[4] K. Cios, W. Pedrycz, R. Swiniarski, and L. Kurgan, *Data Mining - A Knowledge Discovery Approach*. Springer, 2007.

[5] A. Wong, "A statistical parametric and probe yield analysis methodology", *Defect and Fault Tolerance in VLSI Systems*, pp. 131–139, 1996.

[6] M. Sharma, C. Schuermyer, and B. Benware, "Determination of Dominant- Yield-Loss Mechanism with Volume Diagnosis", *IEEE Design & Test of Computers*, vol. 27, no. 3, pp. 54–61, 2010.

[7] D. Drmanac, L. C. Wang, and M. Laisne, "Wafer probe test cost reduction of an RF/A device by automatic testset minimization - A case study", *International Test Conference*, 10 pages, 2011.

[8] N. Sumikawa, J. Tikkanen, L. C. Wang, L. Winemberg, and M. S. Abadir, "Screening Customer Returns with Multivariate Test Analysis", *Proceedings - International Test Conference*, pp. 1–10, 2012.

[9] N. Sumikawa, D. Drmanac, L. C. Wang, L. Winemberg, and M. S. Abadir, "Important test selection for screening potential customer returns", *VLSI-DAT 2011*, pp. 171–174, 2011.

[10] A. Nahar, R. Daasch, and S. Subramaniam, "Burn-in reduction using principal component analysis", *IEEE International Test Conference*, pp. 146–155, 2005.

[11] N. Sumikawa, L.-C. Wang, and M. S. Abadir, "An experiment of burn-in time reduction based on parametric test analysis", *IEEE International Test Conference*, 2012.

[12] P. M. O'Neill, "Production Multivariate Outlier Detection Using Principal Components", *IEEE International Test Conference*, 2008.

[13] A. Nahar, K. M. Butler, J. M. C. Jr, and C. Weinberger, "Statistical Outlier Method Applications", *D3T workshop at ITC*, 2009.

[14] N. Sumikawa, L. C. Wang, and M. S. Abadir, "A pattern mining framework for inter-wafer abnormality analysis", *International Test Conference*, 10 pages, 2013.

[15] C. K. Hsu, F. Lin, K. T. Cheng, W. Zhang, X. Li, J. M. Carulli, and K. M. Butler, "Test data analytics - Exploring spatial and test-item correlations in production test data", *International Test Conference*, pp. 1–10, 2013.

[16] J. Tikkanen, N. Sumikawa, L. C. Wang, and M. S. Abadir, "Multivariate outlier modeling for capturing customer returns - How simple it can be", *IEEE International On-Line Testing Symposium*, pp. 164–169, 2014.

[17] W. R. Daasch, C. G. Shirley, and A. Nahar, "Statistics in semiconductor test: Going beyond yield", *IEEE Design and Test of Computers*, vol. 26, no. 5, pp. 64–73, 2009.

[18] C.-F. Chien, W.-C. Wang, and J.-C. Cheng, "Data mining for yield enhancement in semiconductor manufacturing and an empirical study", *Expert Systems with Applications*, vol. 33, no. 1, pp. 192–198, 2007.

[19] W. C. Tam, O. Poku, and R. D. Blanton, "Systematic defect identification through layout snippet clustering", *International Test Conference*, 2010.

[20] S. learn developers, *Density Estimation*. [Online]. Available: http://scikit-learn.org/stable/modules/density.html (visited on 04/08/2017).

[21] M. P. L. Ooi, Z. A. Kassim, and S. N. Demidenko, "Shortening burn-in test: Application of HVST and weibull statistical analysis", *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 3, pp. 990–999, 2007.

[22] D. M. Hawkins, *Identification of Outliers*, 1st ed. Springer Netherlands, 1980.

[23] C. C. Aggarwal, *Outlier Analysis*, 1st ed. Springer-Verlag New York, 2013.

[24] J. Tikkanen, N. Sumikawa, L. C. Wang, L. Winemberg, and M. S. Abadir, "Statistical outlier screening for latent defects", *IEEE International Reliability Physics Symposium*, pp. 5–8, 2013.

[25] N. Sumikawa, D. Drmanac, L. C. Wang, L. Winemberg, and M. S. Abadir, "Understanding customer returns from a test perspective", *IEEE VLSI Test Symposium*, pp. 2–7, 2011.

[26] H.-M. S. Chang, K.-T. T. Cheng, W. Zhang, X. Li, and K. M. Butler, "Test cost reduction through performance prediction using virtual probe", *International Test Conference*, pp. 1–9, 2011.

[27] K. M. Butler, S. Subramaniam, A. Nahar, J. M. C. Jr., T. J. Anderson, and W. R. Daasch, "Successful Development And Implementation Of Statistical Outlier Techniques On 90nm And 65nm Process Driver Devices", *IEEE IRPS*, pp. 552–559, 2006.

[28] A. Ahmadi, H.-G. Stratigopoulos, K. Huang, A. Nahar, B. Orr, M. Pas, J. M. Carulli, and Y. Makris, "Yield Forecasting Across Semiconductor Fabrication Plants and Design Generations", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[29] T. Ishida, I. Nitta, K. Banno, and Y. Kanazawa, "A volume diagnosis method for identifying systematic faults in lower-yield wafer occurring during mass production", *Asia and South Pacific Design Automation Conference, ASP-DAC*, vol. Di, pp. 670–675, 2014.

[30] C. Knepfler, D. Lammers, and J. O. Nan, "Data analytics: Finding what matters", *Nanochip Fab Solutions*, no. 2, 2014.

[31] S. J. Meyer, "Using Big Data in Manufacturing at Intel's Smart Factories", *White Paper*, no. April, 2016.

[32] Oracle, "Improving Manufacturing Performance with Big Data Architect's Guide and Reference Architecture Introduction", *Oracle corporation*, no. April, 2015.

[33] D. Park, "The Quest for the Quality of Things: Can the Internet of Things deliver a promise of the quality of things?", *IEEE Consumer Electronics Magazine*, vol. 5, no. 2, pp. 35–37, 2016.

[34] H. Matsuhashi, W. Xie, L. Hong, H. Lo, D. Bailey, P. Fernandez, N. Akiya, and J. Jensen, "Online Deployment of Robust Metrology Prediction Model", *AEC/APC Symposium Asia*, pp. 1–2, 2009.

[35] H. Tang, S. Manish, J. Rajski, M. Keim, and B. Benware, "Analyzing volume diagnosis results with statistical learning for yield improvement", *IEEE European Test Symposium, ETS*, 2007.

[36] M. Sharma, B. Benware, L. Ling, D. Abercrombie, L. Lee, M. Keim, H. Tang, W. T. Cheng, T. P. Tai, Y. J. Chang, R. Lin, and A. Man, "Efficiently performing yield

enhancements by identifying dominant physical root cause from test fail data", *International Test Conference*, 2008.

[37] Automotive Electronics Council, "Guidelines for Part Average Testing", *AEC-Q001*, vol. Rev-D, no. December 9, 2011, 2011.

[38] M. J. Moreno-Lizaranzu and F. Cuesta, "Improving electronic sensor reliability by robust outlier screening.", *Sensors*, vol. 13, no. 10, pp. 13 521–13 542, 2013.

[39] W. R. Daasch, J McNames, D Bockelman, and K Cota, "Variance reduction using wafer patterns in IDDQ data", *IEEE International Test Conference*, pp. 189–198, 2000.

[40] W. R. Daasch, K. Cota, J. McNames, and R. Madge, "Neighbor Selection for Variance Reduction in IDDQ and Other Parametric Data", *International Test Conference*, pp. 92–100, 2001.

[41] A. Nahar, K. M. Butler, J. M. C. Jr., and C. Weinberger, "Quality Improvement and Cost Reduction Using Statistical Outlier Methods", *ICCD*, pp. 64–69, 2009.

[42] P. J. Rousseeuw and B. C. van Zomeren, "Unmasking multivariate outliers and leverage points.", *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 633–639, 1990.

[43] N. Sumikawa, D. G. Drmanac, L.-C. Wang, L. R. Winemberg, and M. S. Abadir, "Forward prediction based on wafer sort data - A case study", *International Test Conference*, pp. 1–10, 2011.

[44] W. R. Daasch, "Third- and Fourth-Generation Test Data Analytics", *ITC*, 2015.

[45] A. Rényi, "On measures of dependence", *Acta Mathematica Academiae Scientiarum Hungaricae*, vol. 10, no. 3-4, pp. 441–451, 1959.

[46] J. Jacod and P. Protter, *Probability Essentials*. 2000.

[47] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: an overview with application to learning methods.", *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[48] J Shawe-Taylor and N Cristianini, *Kernel Methods for Pattern Analysis*. 2004.

[49] F. R. Bach and M. I. Jordan, "Kernel Independent Component Analysis", *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[50] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines", *J. Mach. Learn. Res.*, vol. 2, pp. 67–93, 2001.

[51] A. Gretton, R. Herbrich, A. J. Smola, O. Bousquet, and B. Schölkopf, "Kernel Methods for Measuring Independence", *Journal of Machine Learning Research*, vol. 6, pp. 2075–2129, 2005.

[52] M. Kuss and T. Graepel, "The Geometry of Kernel Canonical Correlation Analysis", Tech. Rep. Technical Report 108, 2003, May.

[53] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[54] Kane V, *Process Capability Indices_1986*, 1986.

[55] J. Tikkanen, S. Siatkowski, N. Sumikawa, L.-c. Wang, and M. S. Abadir, "Yield Optimization Using Advanced Statistical Correlation Methods", *International Test Conference*, 2014.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2012.

[57] R. Agrawal, D. Gunopulos, and F. Leymann, "Mining Process Models from Workflow Logs", *International Conference on Extending Database Technology*, 1998.

[58] W. M. P. van der Aalst, A. J.M. M. Weijters, and L Maruster, "Workflow Mining: Discovering process models from event logs", *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, 2004.

[59] D. Angluin and C. H. Smith, "Inductive Inference: Theory and Methods", *ACM Computing Surveys*, vol. 15, no. 3, pp. 237–269, 1983.

[60] W. van der Aalst, B. van Dongen, C. Günther, M. R.S., A. de Medeiros, R. A.K., R. A., S. V., H. M. Verbeek, and A. Weijters, "ProM 4.0: comprehensive support for real process analysis", *In: Kleijn J., Yakovlev A. (eds) Petri Nets and Other Models of Concurrency – ICATPN 2007.*, no. Lecture Notes in Computer Science, vol 4546, 2007.

[61] W. M. P. van der Aalst, V. Rubin, H. M. W. Verbeek, B. F. van Dongen, E. Kindler, and C. W. Günther, "Process mining: A two-step approach to balance between underfitting and overfitting", *Software and Systems Modeling*, vol. 9, no. 1, pp. 87–111, 2010.

[62] Ian Goodfellow, Y. Bengio, and A. Courville, "Deep learning", *MIT Press*, no. http://www.deeplearningbook.org, 2016.

[63] FederalCommunicationsCommission, "Operation of Radar Services in the 76-81 GHz Band", vol. FCC-15-16, 2015. [Online]. Available: https://www.fcc.gov/document/operation-radar-services-76-81-ghz-band.

[64] F. E. Grubbs, "Procedures for Detecting Outlying Observations in Samples", *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[65] F Hernandez and R. Johnson, "The large-sample behavior of transformations to normality", *Journal of American Statistical Association*, vol. 75, no. 372, pp. 855–861, 1980.

[66] Y.-M. Chou, A. M. Polansky, and R. L. Mason, "Transforming non-normal data to normality in statistical process control", *Journal of Quality Technology*, vol. 30, no. 2, pp. 133–141, 1998.

[67] B. Silverman, "Density estimation for statistics and data analysis", *Chapman and Hall*, vol. 37, no. 1, pp. 1–22, 1986.

[68] S. Siatkowski, C.-L. Chang, L.-C. Wang, N. Sumikawa, L. Winemberg, and W. R. Daasch, "Generalization of an outlier model into a "global" perspective", *International Test Conference*, 2015.

[69] R. W. Daasch and R. Madge, "Variance reduction and outliers: statistical analysis of semiconductor test data", *ITC*, 2005.

[70] J. M. Buhmann, "Information theoretic model validation for clustering", *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 1398 –1402, 2010.

[71] S. Maneewongvatana and D. M. Mount, "On the Efficiency of Nearest Neighbor Searching with Data Clustered in Lower Dimensions", *Computational Science - ICCS 2001*, pp. 842–851, 2001.