Automatic Clustering of Wafer Spatial Signatures

Wangyang Zhang[1], Xin Li[1], Sharad Saxena[2], Andrzej Strojwas[1], Rob Rutenbar[3]
[1]ECE Department, Carnegie Mellon University, Pittsburgh, PA 15213
[2]PDF Solutions, 101 Renner Trail, Richardson, TX 75082
[3]CS Department, University of Illinois at Urbana-Champaign, Urbana IL 61801
{wyzhang, xinli, ajs}@ece.cmu.edu, sharad.saxena@pdf.com, rutenbar@illinois.edu

ABSTRACT

In this paper, we propose a methodology based on unsupervised learning for automatic clustering of wafer spatial signatures to aid yield improvement. Our proposed methodology is based on three steps. First, we apply sparse regression to automatically capture wafer spatial signatures by a small number of features. Next, we apply an unsupervised hierarchical clustering algorithm to divide wafers into a few clusters where all wafers within the same cluster are similar. Finally, we develop a modified L-method to determine the appropriate number of clusters from the hierarchical clustering result. The accuracy of the proposed methodology is demonstrated by several industrial data sets of silicon measurements.

1. INTRODUCTION

With the continued scaling of CMOS technology, process variation has become a critical issue for design and manufacture of integrated circuits [1]. Large-scale performance variability has been observed for integrated circuits at advanced technology nodes, resulting in significant yield loss. For this reason, reducing process variation to improve parametric yield is an extremely important task that is carried out throughout the lifecycle of any process and product.

In order to rapidly improve parametric yield, it is important to identify the key factors that significantly contribute to the yield loss [2]. To monitor the process characteristics, a number of test structures, such as ring oscillators [3] and transistor arrays [4], are placed within each chip or in the scribe line. An important observation is that different wafers may exhibit substantially different spatial signatures for the measurements of these test structures [5]. Different spatial signatures suggest that the underlying variation sources can be different for these wafers and, therefore, can be used to reveal a large number of yield-limiting factors, such as process shift/drift, mismatch between equipments, etc. If we can capture the spatial signature of each wafer with an accurate model, and further automatically partition all wafers into different groups based on such spatial signatures where each group carries a similar spatial signature, it would provide important insights to help process engineers for yield improvement. In particular, process engineers can rely on the information to prioritize different yield improvement strategies and focus on the variation sources associated with significant yield loss.

The problem of automatically grouping wafers with similar spatial signatures can be defined as a *clustering analysis* problem in statistics. While clustering analysis has been extensively studied in the statistics community, a number of unique characteristics of our wafer clustering problem must be carefully considered in order to obtain accurate clustering results:

Large random variation: the performance measurements collected from test structures may be subject to large-scale random variation. As random variation becomes increasingly large with technology scaling [12], it obscures the spatial signature, thereby making the spatial signature non-trivial to identify.

Missing and outlier measurements: Defects in the manufacturing process, as well as measurement errors, may generate missing measurements. In this case, no data may be collected from a number of test structures, or outlier measurements collected from these test structures may significantly deviate from the regular variation range [7]. Meaningful clustering results cannot be obtained if the missing or outliner measurements are not properly handled.

Abnormal wafers: Because of equipment malfunction, there can be a small number of abnormal wafers whose spatial signatures are substantially different from the others [5]. We would like to automatically detect these abnormal wafers, rather than merging them into the main clusters. It, in turn, poses a unique challenge to the clustering algorithm, as will be explained in detail in Section 3.

Unknown number of clusters: Most clustering algorithms require knowing the number of clusters, or having user-defined parameters related to the number of clusters. In our wafer clustering application, the number of clusters cannot be known in advance. Therefore, additional efforts must be made to optimally determine the number of clusters from the measurement data.

Based on the aforementioned characteristics, we propose a new methodology for automatic clustering of wafer spatial signatures. Our proposed method consists of three steps. First, robust feature extraction based on sparse regression is performed on the measurement data, representing the spatial signature of each wafer by a small number of features. The impact of random variation is greatly reduced in our proposed feature space and, furthermore, the proposed feature extraction is extremely robust to missing and outlier measurements. Next, a clustering algorithm is performed on the extracted features. Since the number of clusters is not known in advance, the clustering algorithm does not directly generate the final clustering result. Instead, a set of possible clustering results are generated according to different settings of the clustering algorithm. Finally, a cluster selection algorithm is applied to automatically choose the optimal clustering result that best explains the data.

The remainder of the paper is organized as follows. In Section 2 we present our robust feature extraction algorithm, and then describe the clustering algorithm in Section 3. The algorithm for optimal cluster selection is presented in Section 4. The efficacy of our proposed method is demonstrated by several industrial examples in Section 5. Finally, we conclude in Section 6.

DAC'13, May 29 - June 07 2013, Austin, TX, USA.
Copyright 2013 ACM 978-1-4503-2071-9/13/05...$15.00

2. ROBUST FEATURE EXTRACTION

The goal of robust feature extraction is to represent the spatial signature of each wafer by a small number of features that minimize the impact of random variation, missing data and outlier measurements. We represent the parametric metric (e.g., ring oscillator frequency, leakage current, etc) measured from L wafers as a set of two-dimensional functions: $\{b_{(l)}(x, y);\ l = 1, 2, \ldots, L\}$, where l denotes the wafer label, and $x \in \{1, 2, ..., P\}$ and $y \in \{1, 2, ..., Q\}$ denote the spatial coordinates on the wafer. Each spatial variation function $b_{(l)}(x, y)$ contains two different components:

$$b_{(l)}(x, y) = s_{(l)}(x, y) + r_{(l)}(x, y) \quad (l = 1, 2, \cdots, L), \tag{1}$$

where $\{s_{(l)}(x, y);\ x = 1, 2, ..., P;\ y = 1, 2, ..., Q\}$ and $\{r_{(l)}(x, y);\ x = 1, 2, ..., P;\ y = 1, 2, ..., Q\}$ stand for the spatially correlated component and the uncorrelated random component, respectively. In order to reduce the impact of random variation, we would like to represent the spatial signature of each wafer by using its spatially correlated component only. Specifically, if the spatially correlated variation is modeled by the linear combination of λ basis functions:

$$b_{(l)}(x, y) = \sum_{j=1}^{\lambda} \eta_{(l),j} \cdot A_j(x, y) + r_{(l)}(x, y) \quad , \tag{2}$$

we define the features of the lth wafer as the following vector:

$$\eta_{(l)} = \begin{bmatrix} \eta_{(l),1} & \eta_{(l),2} & \cdots & \eta_{(l),\lambda} \end{bmatrix}^T . \tag{3}$$

By using the λ features in (3) to represent the spatial signature of the lth wafer, the uncorrelated random variation $r_{(l)}(x, y)$ would not impact the subsequent clustering process, and the clustering result would be made insensitive to random variation.

In practice, we do not know the wafer spatial signatures in advance. Therefore, we adopt the sparse regression idea in [8] to automatically select an appropriate set of basis functions from a dictionary that covers all possible spatial patterns. Such a dictionary can be customized by process engineers based on their prior knowledge. In the worst scenario, if no prior knowledge is available, a general dictionary containing Discrete Cosine Transform (DCT) [14] functions can be applied. The DCT functions are defined as:

$$A_{u,v}(x, y) = \alpha_u \cdot \beta_v \cdot \cos\frac{\pi(2x-1)(u-1)}{2 \cdot P} \cdot \cos\frac{\pi(2y-1)(v-1)}{2 \cdot Q} , \tag{4}$$
$$(u = 1, 2, \cdots, P;\ v = 1, 2, \cdots, Q)$$

where

$$\alpha_u = \begin{cases} \sqrt{1/P} & (u = 1) \\ \sqrt{2/P} & (2 \le u \le P) \end{cases} \tag{5}$$

$$\beta_v = \begin{cases} \sqrt{1/Q} & (v = 1) \\ \sqrt{2/Q} & (2 \le v \le Q) \end{cases} . \tag{6}$$

The DCT coefficients, denoted as {□$_{(l)}(u, v)$; $u = 1, 2, ..., P$; $v = 1, 2, ..., Q$}, represent the frequency-domain components of the spatial variation function $\{b_{(l)}(x, y);\ x = 1, 2, ..., P;\ y = 1, 2, ..., Q\}$. An important property of DCT is that if the spatial variation $b_{(l)}(x, y)$ exhibits a spatially correlated pattern, a vast majority of the DCT coefficients are close to zero. This unique property of sparseness has been observed in many image processing tasks and serves as the foundation of the compression algorithm for JPEG [14]. It has been recently explored by several works in the literature to model wafer-level spatial variation [6]-[8]. On the other hand, uncorrelated random variation can be characterized as white noise [8] and evenly distributed over all frequencies. Therefore, the corresponding DCT coefficients are relatively small. It, in turn, implies that spatially correlated variation can be accurately represented by a small number of dominant DCT coefficients.

After selecting the dictionary of basis functions, the following sparse regression [8] is formulated to generate the features:

$$\begin{aligned} \underset{\eta_{(l)}}{\text{minimize}} \quad & \left\| A_{(l)} \cdot \eta_{(l)} - B_{(l)} \right\|_2^2 \\ \text{subject to} \quad & \left\| \eta_{(l)} \right\|_0 \le \lambda \end{aligned} , \tag{7}$$

where $||\bullet||_2$ and $||\bullet||_0$ stand for the L_2-norm (i.e., the square root of the summation of the squares of all elements) and the L_0-norm (i.e., the number of non-zero elements) of a vector respectively,

$$B_{(l)} = \begin{bmatrix} b_{(l)}\left(x_{(l),1}, y_{(l),1}\right) & \cdots & b_{(l)}\left(x_{(l),N_{(l)}}, y_{(l),N_{(l)}}\right) \end{bmatrix}^T \tag{8}$$

represents the measurement data collected from $N_{(l)}$ different spatial locations $\{(x_{(l),i}, y_{(l),i});\ i = 1, 2, ..., N_{(l)}\}$ of the lth wafer,

$$\eta_{(l)} = \begin{bmatrix} \eta_{(l),1} & \cdots & \eta_{(l),M} \end{bmatrix}^T \tag{9}$$

represents the unknown coefficients corresponding to the M basis functions in the dictionary. If the DCT dictionary is applied, the total number of basis functions (i.e., M) is equal to PQ. The matrix $A_{(l)}$ in (7) is $N_{(l)}$-by-M and is defined as:

$$A_{(l)} = \begin{bmatrix} A_{(l),1,1} & A_{(l),1,2} & \cdots & A_{(l),1,M} \\ A_{(l),2,1} & A_{(l),2,2} & \cdots & A_{(l),2,M} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(l),N_{(l)},1} & A_{(l),N_{(l)},2} & \cdots & A_{(l),N_{(l)},M} \end{bmatrix} , \tag{10}$$

where $A_{(l),i,j}$ corresponds to the value of the jth basis function for the ith measurement on the lth wafer. The optimization in (7) attempts to use a small number of (i.e., λ) dominant basis functions to approximate the spatially correlated variation of the lth wafer. It can be efficiently solved by the numerical algorithm developed in [8] where the optimal value of λ can be automatically determined using cross-validation. Once the optimization in (7) is solved, the resulting λ dominant coefficients become the features of each wafer in (3). More details on sparse regression can be found in [6]-[8].

The sparse regression approach is extremely robust to missing measurements [6]-[8]. Moreover, it can be further made insensitive to measurement outliers by applying advanced outlier detection techniques [7], [15]. Therefore, by performing sparse regression to extract the important features, the subsequent clustering process can be appropriately shielded from the missing and outlier measurements, as will be discussed in detail in the following sections.

3. CLUSTERING ALGORITHM

After the features describing the wafer signatures are extracted, the next step is to apply a clustering algorithm to partition all wafers into multiple clusters. Many clustering algorithms have been proposed in the statistics community, such as k-means clustering [9], density-based clustering [10] and hierarchical clustering [13]. Each algorithm is based upon a specific assumption of the data and not all these algorithms are suitable for our wafer clustering application.

The traditional k-means method partitions the data into K clusters that minimize the following cost function:

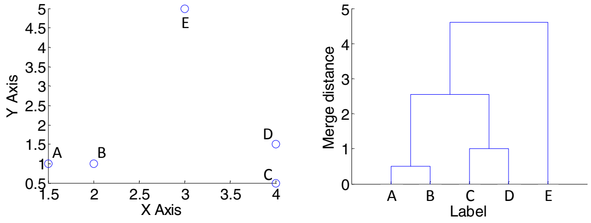
$$\sum_{i=1}^{K} \sum_{l \in c_i} \left\| \eta_{(l)} - \mu_i \right\|_2^2 , \tag{11}$$

where c_i denotes the index set of the wafers belonging to the ith cluster, $\eta_{(l)}$ is the feature vector of the lth wafer, and μ_i is the centroid of the ith cluster:

$$\mu_i = \frac{1}{|c_i|} \cdot \sum_{l \in c_i} \eta_{(l)} \ , \tag{12}$$

where $|c_i|$ stands for the size of the set c_i. The number of clusters (i.e., K) is a parameter that must be specified by the user in advance. The clustering result can be efficiently found by the Expectation-Maximization (EM) algorithm [13] developed by the statistics community.

However, for our wafer clustering application, an important problem that prevents k-means from achieving accurate results is the existence of abnormal wafers. Abnormal wafers are a small number of wafers whose spatial signatures are substantially different from any of the main clusters, typically because of equipment malfunction. An ideal clustering algorithm should result in a number of separate clusters with very small sizes to reflect the abnormal wafers, rather than merging them into the main clusters. However, detecting such small clusters is often not possible with the k-means clustering algorithm [13], as will be demonstrated by our experimental results in Section 5.



(a) (b)

Figure 1. (a) A synthetic two-dimensional data set with 5 points {A, B, C, D, E}. (b) The dendrogram generated by hierarchical clustering.

An alternative algorithm that does not suffer from the aforementioned problem is hierarchical clustering [13]. Unlike the k-means method which explicitly minimizes a cost function, hierarchical clustering builds clusters in a greedy manner. Suppose that there are N data points in total. Hierarchical clustering first assigns an individual cluster for each point. Next, N–1 merging steps are performed iteratively, where two clusters that are closest in distance are merged in each step. Namely, the data points that are close will be merged first, and those that are far away will not be merged until the end of the iteration process.

To intuitively explain the idea of hierarchical clustering, we construct a synthetic data set with 5 points shown in Figure 1(a). When hierarchical clustering is applied, the first iteration step merges the data points A with B and the second iteration step merges the data points C with D. Next, the two clusters containing {A, B} and {C, D} respectively are merged in the third iteration step. Finally, the data point E is merged with {A, B, C, D} in the last iteration step. The clustering result can be represented as a *dendrogram* shown in Figure 1(b).

Studying Figure 1(b) reveals an important fact that the height of each node reflects its *merge distance*, meaning the distance of two clusters that are merged to form this node. Note that close data points should be merged early, while distant data points will not be merged until the end. However, hierarchical clustering does not directly generate the final clustering result (i.e. the cluster labels for each data point). We will further discuss the algorithm to determine the cluster labels in Section 4.

An important component that must be determined for hierarchical clustering is how to define the distance between clusters. While the distance between two individual data points can be simply defined by their Euclidean distance:

$$dist\left(\eta_{(l)}, \eta_{(k)}\right) = \left\| \eta_{(l)} - \eta_{(k)} \right\|_2 \ , \tag{13}$$

the definition of distance between two clusters containing multiple data points is not unique. Different definitions of cluster distance have been proposed, and each of them corresponds to a different assumption about the cluster structure. We need to select the distance definition that best matches our goal for wafer clustering. In this work, the following definition is used, which calculates the distance between two clusters as the maximal distance between any two data points in the clusters:

$$dist\left(c_l, c_k\right) = \sup_{i \in c_l, j \in c_k} \left\| \eta_{(i)} - \eta_{(j)} \right\|_2 \ , \tag{14}$$

where sup(•) denotes the supremum (i.e., the least upper bound) of a set. The hierarchical clustering algorithm based on the distance metric in (14) is referred to as *complete-link hierarchical clustering* [13]. The physical meaning of (14) is that a cluster will be formed if and only if *all* members in the cluster are completely connected, i.e. within a small distance to each other. This definition matches our goal for wafer clustering: since all wafers in the same cluster should carry the same spatial signature, we want these wafers to be similar to each other.

To further explain why Eq. (14) is an appropriate choice for our application, we compare it with another commonly used definition based on minimal distance:

$$dist\left(c_l, c_k\right) = \inf_{i \in c_l, j \in c_k} \left\| \eta_{(i)} - \eta_{(j)} \right\|_2 \ , \tag{15}$$

where inf(•) denotes the infimum (i.e., the greatest lower bound) of a set. The hierarchical clustering algorithm based on the distance metric in (15) is referred to as *single-link hierarchical clustering* [13]. The assumption behind single-link hierarchical clustering is that two data points should belong to the same cluster, as long as there exists a path connecting these two data points such that any adjacent pair of points along this path is close in distance. As a result, single-link hierarchical clustering often generates elongated clusters, where distant data points are connected by a long path in between. While this type of cluster is suitable for many practical applications, it is undesirable for our wafer clustering. For instance, the change in process condition may not occur abruptly during the manufacturing process, but gradually drift from one state to another. Such a drift can happen because of, for example, equipment aging [16]. By employing single-link hierarchical clustering, we are unable to split the wafers into different clusters to reflect the drift of process condition. Note that several other clustering techniques, e.g., density-based clustering [10], are also based on the idea of forming a connecting path to define clusters. Therefore, they are not suitable for our wafer clustering application either.

In summary, complete-link hierarchical clustering can naturally break down a long string of data points into small clusters and, therefore, it is most suitable for our application in this paper. In Section 5, we will show several examples where the correct clusters detected by complete-link hierarchical clustering cannot be found by either single-link hierarchical clustering or k-means clustering.

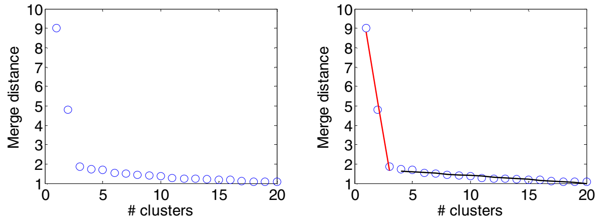
4. CLUSTER SELECTION

In the previous section, we propose to apply complete-link hierarchical clustering for our application. However, as previously discussed, hierarchical clustering does not directly generate the cluster labels. To achieve automatic clustering and minimize human efforts, a cluster selection algorithm must be developed to automatically choose the appropriate cluster labels from the hierarchical clustering result.

The traditional approach to select the clusters from the hierarchical clustering result is based on the *inconsistency coefficient method* [17]. It visits each node in the dendrogram and compares its merge distance with the average merge distance of all nodes below it. The difference is quantitatively defined by the following inconsistency coefficient:

$$I_k = \frac{d_k - \mu_k}{\sigma_k}, \tag{16}$$

where I_k represents the inconsistency coefficient of the kth node, d_k is the merge distance of the kth node, μ_k is the average merge distance of the kth node and all nodes below it, and σ_k is the standard deviation of the merge distances of the kth node and all nodes below it. The nodes with inconsistency coefficient higher than a user-defined threshold are broken, yielding distinct clusters. This threshold value is often empirically assigned, and its optimal value can vary significantly over different applications or even different data sets. On the other hand, the clustering result is extremely sensitive to the aforementioned threshold value. Hence, it is non-trivial to develop a fully automatic clustering process based on the inconsistency coefficient method.



(a) (b)

Figure 2. (a) The error curve of complete-link hierarchical clustering for a synthetic data set. (b) The optimal number of clusters can be found by fitting the curve with two lines.

An alternative approach to select the number of clusters that has gained popularity in recent years is based on the L-method [11]. The L-method is derived from the fact that for many clustering algorithms, it is possible to plot an error curve where the x-axis is the number of clusters and the y-axis is the evaluation metric internally used by the clustering algorithm. For hierarchical clustering, the evaluation metric of having i clusters is defined as the merge distance of the (N–i)-th merge [11]. While the error curve generally presents a decreasing trend, it typically has a sharp transition at the optimal clustering setup. For example, Figure 2(a) plots the error curve of complete-link hierarchical clustering for a synthetic data set with three clusters. It can be seen that the transition point is at $x = 3$. The L-method attempts to match human intuition by defining the criterion that if we find two consecutive lines that optimally fit the error curve, the intersection point of the two lines determines the transition point of the error curve. For example, Figure 2(b) accurately fits the error curve by two lines where one line fits the data within $x \in [1, 3]$ and the other line fits the data within $x \in [4, 20]$. It, in turn, determines that the data set should be partitioned into 3 clusters.

In what follows, we will first describe the mathematic formulation of the L-method and then discuss its limitation. Consider an error curve such as Figure 2(a) where the value of the x-axis varies from $x = 1$ to $x = B$. We partition the data points into the left and right sequences at $x = c$. The left sequence has the data points with $x \in \{1, ..., c\}$ and the right sequence has the data points with $x \in \{c+1, ..., B\}$. Next, we find two optimal lines that minimize the mean-squares error to fit the left and right parts of the error curve respectively:

$$\underset{a_{lc}, b_{lc}}{\text{minimize}} \quad \left\| y_{lc} - a_{lc} - b_{lc} \cdot x_{lc} \right\|_2^2 \tag{17}$$

$$\underset{a_{rc}, b_{rc}}{\text{minimize}} \quad \left\| y_{rc} - a_{rc} - b_{rc} \cdot x_{rc} \right\|_2^2 , \tag{18}$$

where

$$x_{lc} = \begin{bmatrix} 1 & 2 & \cdots & c \end{bmatrix}^T \tag{19}$$

$$x_{rc} = \begin{bmatrix} c+1 & c+2 & \cdots & B \end{bmatrix}^T , \tag{20}$$

y_{lc} and y_{rc} are the values of the evaluation metric at x_{lc} and x_{rc} respectively. Eq. (17) and (18) can be solved by least-squares fitting, yielding the following root-mean-squared error:

$$RMSE_{lc} = \frac{1}{\sqrt{c}} \cdot \left\| y_{lc} - a_{lc} - b_{lc} \cdot x_{lc} \right\|_2 \tag{21}$$

$$RMSE_{rc} = \frac{1}{\sqrt{B-c}} \cdot \left\| y_{rc} - a_{rc} - b_{rc} \cdot x_{rc} \right\|_2 , \tag{22}$$

where a_{lc} and b_{lc} are the solution of (17), and a_{rc} and b_{rc} are the solution of (18). The total root-mean-squared error associated with the transition point $x = c$ is defined as the weighted sum of $RMSE_{lc}$ and $RMSE_{rc}$:

$$RMSE_c = \frac{c}{B} RMSE_{lc} + \frac{B-c}{B} RMSE_{rc} . \tag{23}$$

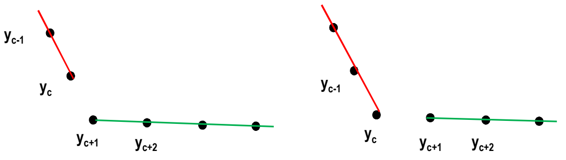
The optimal number of clusters is then defined by selecting the c value that minimizes the total error in (23):

$$\underset{c}{\text{minimize}} \quad RMSE_c . \tag{24}$$

In practice, the number of clusters is often much smaller than the number of data points. Therefore, when directly applying the criterion (24) to the entire data set, a large number of possible c values corresponding to extremely fine-grain clusters are irrelevant and may lead to an inaccurate result due to highly imbalanced left and right sequences. Therefore, the L-method is applied iteratively to the error curve. Starting from solving (24) for the entire error curve, each iteration step reduces the number of data points included in the next iteration to:

$$B_{next} = \max(2 \cdot c, 20) , \tag{25}$$

where c is the optimal number of clusters determined in the current iteration step, and $2 \cdot c$ is the number to keep the left and right sequences balanced. The total number of data points is not permitted to drop below 20, which is an empirical number proposed in [11] in order to keep a reasonable number of data points to fit the lines. The L-method stops when the number of data points does not change over two successive iteration steps.



(a) (b)

Figure 3. An synthetic example where the error can be minimized by either (a) $c = 2$ or (b) $c = 3$.

While the L-method attempts to match human intuition in finding the transition point of the error curve, we notice that its definition of the transition point is counter-intuitive. To explain its limitation, we construct a synthetic example in Figure 3, where the optimal solution with human inspection is c = 3. However, Figure 3(a) and (b) show the results fitted by setting c = 2 and c = 3 respectively. It can be seen that both solutions yield extremely small error. Therefore, the choice of c = 2 or c = 3 by the L-method is arbitrary in this example.

Based on the aforementioned observation, we propose to add a post-processing step to the traditional L-method to accurately determine the number of clusters. The key idea is to detect if a sharp transition occurs at the current point c or the next point c+1. The number of clusters is added by one, if the next point causes a sharp transition in the error curve. Specifically, we propose to use the following quantity to measure the transition rate:

$$s(c)=\left[\log(y_{c+1})-\log(y_c)\right]-\left[\log(y_c)-\log(y_{c-1})\right], \qquad (26)$$

where y_{c-1}, y_c and y_{c+1} are the values of the evaluation metric at $x = c-1$, $x = c$ and $x = c+1$, respectively. The number of clusters is increased by one, if $s(c+1)$ is greater than $s(c)$. Eq. (26) is essentially the second-order difference of the data series log(y). A large second-order difference means a significant change in the slope, thereby indicating an abrupt transition of the error curve. We take the logarithm for the evaluation metric y, because comparing the ratio between two consecutive data points is more intuitive than comparing their absolute difference. We summarize the major steps of the modified L-method for cluster selection in Algorithm 1.

Algorithm 1: Modified L-method for cluster selection

1. Start from a vector $y \in R^B$ representing the value of the evaluation metric associated with $x \in \{1, 2, ..., B\}$.
2. Find the optimal number of clusters (i.e., c) according to the criterion (24).
3. Compare $s(c)$ and $s(c+1)$ defined by (26). If $s_{c+1} > s_c$, then $c = c + 1$.
4. Calculate the value of B_{next} by (25) to determine the number of data points that should be included in the next iteration step.
5. If $B_{next} = B$, stop iteration. Otherwise, set $B = B_{next}$ and go to Step 1.

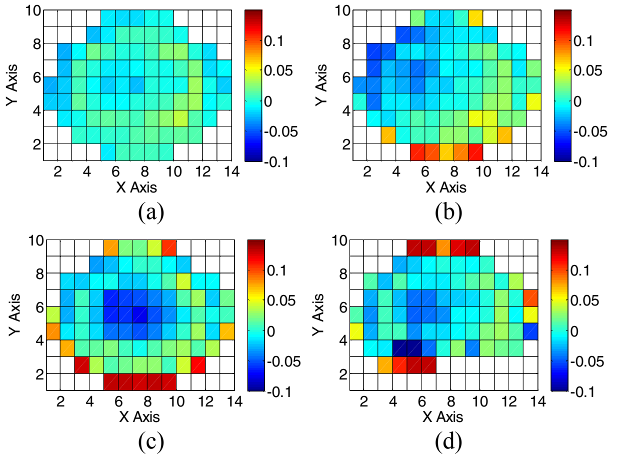
Note that Algorithm 1 is not restricted to hierarchical clustering only. Instead, it can be applied to select the optimal number of clusters for any clustering algorithm where the error curve (i.e., the evaluation metric vs. the number of clusters) can be generated. For instance, Algorithm 1 can be successfully applied to k-means clustering, where the evaluation metric is defined by the cost function in (11).

5. EXPERIMENTAL EXAMPLES

In the previous sections, we have proposed a wafer clustering methodology that mainly consists of three components: (i) robust feature extraction, (ii) complete-link hierarchical clustering, and (iii) cluster selection. In this section, we demonstrate the efficacy of the proposed methodology based on several industrial data sets for a commercial CMOS process below 90nm.

First, we consider the measurement data of NMOS drain saturation current (I_{dsat}) collected by the scribe-line test structures from 69 wafers. We apply the proposed methodology to cluster these wafers. In this example, four clusters, referred as Cluster "a", "b", "c" and "d", are identified where the numbers of wafers belonging to these four clusters are 34, 23, 9 and 3, respectively. Figure 4 shows the averaged wafer map for the four clusters. It can be seen that these clusters indeed contain distinct spatial signatures. In particular, the wafers in Cluster "a" do not carry significant spatially correlated variation. The wafers in Cluster "b" present a strong edge effect and an increasing trend from the top-left corner to the bottom-right corner. The wafers in Cluster "c" have strong edge and center effects. Finally, the wafers in Cluster "d" have a large number of missing measurements at the bottom of the wafer.

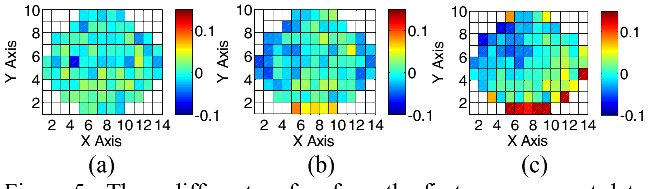
In this example, the aforementioned spatial signatures cannot be accurately detected by k-means clustering or single-link hierarchical clustering. K-means clustering only detects three clusters where Cluster "b" and "c" in Figure 4 are merged into a single cluster. Therefore, it fails to detect the different spatial signatures presented by these two clusters. On the other hand, if single-link hierarchical clustering is applied, Cluster "a", "b" and "c" in Figure 4 are merged into one cluster. The fundamental reason is that there does not exist a clear boundary between these three clusters. To further validate this reason, we plot three different wafer maps in our data set in Figure 5. It can be seen that while there exist substantially different spatial signatures between Figure 5(a) and Figure 5(c), Figure 5(b) has a spatial signature that is similar to both Figure 5(a) and Figure 5(c). This observation may occur because of, for example, process drift. In this case, single-link hierarchical clustering will merge Figure 5(a) and Figure 5(c) into the same cluster because they are connected by Figure 5(b). Complete-link hierarchical clustering, however, requires all wafers in the same cluster to be similar and, therefore, does not suffer from this issue. Finally, we also verify that no satisfactory clustering result can be generated, if the inconsistency coefficient method is applied for cluster selection.



(a) (b)

(c) (d)

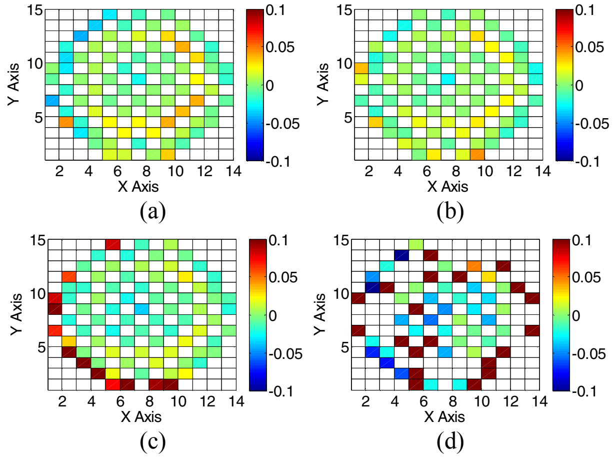
Figure 4. Averaged wafer maps (normalized) of four different clusters detected by the proposed methodology for the first measurement data set of drain saturation current (I_{dsat}).



(a) (b) (c)

Figure 5. Three different wafers from the first measurement data set of drain saturation current (I_{dsat}).

Next, we further consider the I_{dsat} measurements from another data set with 82 wafers. In this data set, not all test structures are measured; instead, they are sampled in a "checkerboard" style to reduce the test cost. In this example, the proposed methodology again generates four clusters, referred to as Cluster “a”, “b”, “c” and “d”. The numbers of wafers belonging to these four clusters are 43, 18, 20 and 1, respectively. Figure 6 shows the averaged wafer map for these four clusters. Inspecting Figure 6, it can be seen that although Cluster “b” presents larger spatially correlated variation than Cluster “a”, the difference in spatial signature between these two clusters is not significant. Therefore, they can be simply merged into one cluster after manually inspecting Figure 6(a) and Figure 6(b). Note that even though the clustering result does not exactly match human intuition in this example, the aforementioned manual inspection requires little human effort. Cluster “c” and “d” detected by the proposed method carry completely different spatial signatures compared to Cluster “a” and “b”. Namely, the wafers in Cluster “c” have a significant edge effect at the bottom-left corner of the wafer, and Cluster “d” contains an abnormal wafer with a completely different spatial signature compared to other wafers.



(a)

(b)

(c)

(d)

Figure 6. Averaged wafer maps (normalized) of four different clusters detected by the proposed methodology for the second measurement data set of drain saturation current (I_{dsat}).

In this example, applying k-means clustering or single-link hierarchical clustering fails to detect all the distinct signatures in Figure 6. In particular, k-means clustering generates two clusters where Cluster “a”, “b” and “d” in Figure 6 are merged into one cluster and Cluster “c” in Figure 6 forms a separate cluster. While the k-means method merges Cluster “a” and “b”, the abnormal wafer is also merged into this large cluster and cannot be detected by simple inspection. On the other hand, single-link hierarchical clustering generates two clusters where Cluster “a”, “b” and “c” in Figure 6 are merged into one cluster and Cluster “d” in Figure 6 forms a separate cluster. Therefore, it fails to detect the wafers with edge effect. For these reasons, the proposed methodology with complete-link hierarchical clustering provides the best accuracy in this example.

6. CONCLUSIONS

In this paper, we develop an accurate three-step methodology for automatic clustering of wafer spatial signatures. First, the spatial signatures are automatically captured by a small number of features based on sparse regression. Second, complete-link hierarchical clustering is performed on the extracted features. Finally, a modified L-method is performed on the hierarchical clustering result for cluster selection. The efficacy of the proposed methodology, as well as its superior accuracy over other alternative approaches, has been demonstrated by a number of industrial data sets of silicon measurements. In our future work, we will further apply the clustering results to identify the critical factors for yield improvement.

7. ACKNOWLEDGEMENTS

This work has been supported in part by the National Science Foundation.

8. REFERENCES

[1] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2011.

[2] N. Kupp, M. Slamani and Y. Makris, “Correlating inline data with final test outcomes in analog/RF devices,” *IEEE DATE*, 2011.

[3] M. Bhushan, A. Gattiker, M. Ketchen and K. Das, “Ring oscillators for CMOS process tuning and variability control,” *IEEE Trans. Semiconductor Manufacturing*, vol. 19, no. 1, pp. 10-18, Feb. 2006.

[4] S. Saxena, C. Hess, H. Karbasi, A. Rossoni, S. Tonello, P. McNamara, S. Lucherini, S. Minehane, C. Dolainsky and M. Quarantelli “Variation in transistor performance and leakage in nanometer-scale technologies,” *IEEE Trans. Electron Devices*, vol. 55, pp. 131-144, Jan. 2008.

[5] A. Strojwas, “Conquering process variability: A key enabler for profitable manufacturing in advanced technology nodes,” *IEEE International Symposium on Semiconductor Manufacturing*, pp. xxiii-xxxii, 2006.

[6] X. Li, R. Rutenbar and R. Blanton, “Virtual probe: a statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits,” *IEEE ICCAD*, pp. 433-440, 2009.

[7] W. Zhang, X. Li, E. Acar, F. Liu and R. Rutenbar, “Multi-wafer virtual probe: minimum-cost variation characterization by exploring wafer-to-wafer correlation,” *IEEE ICCAD*, pp. 47-54, 2010.

[8] W. Zhang, K. Balakrishnan, X. Li, D. Boning and R. Rutenbar, “Toward efficient spatial variation decomposition via sparse regression,” *IEEE ICCAD*, pp. 162-169, 2011.

[9] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, 1967.

[10] M. Ester, H. Kriegel, J. Sander and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” *International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[11] S. Salvador and P. Chan, “Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms,” *International Conference on Tools with AI*, pp. 576-584, 2004.

[12] M. Orshansky, S. Nassif and D. Boning, *Design for Manufacturability and Statistical Design: A Constructive Approach*, Springer, 2007.

[13] P. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*. Addison-Wesley, 2006.

[14] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice Hall, 2007.

[15] R. Maronna, R. Martin and V. Yohai, *Robust Statistics: Theory and Methods*, John Wiley and Sons, 2006.

[16] G. May and C. Spanos, *Fundamentals of Semiconductor Manufacturing and Process Control*, Wiley-IEEE Press, 2006.

[17] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.