Chapter 2
Diagnosis Using Support Vector Machines (SVM)

Diagnosis of functional failures at the board level is critical for improving product yield and reducing manufacturing cost. State-of-the-art board-level diagnostic software is unable to cope with high complexity and ever-increasing clock frequencies, and the identification of the root cause of failure on a board is a major problem today. Ambiguous or incorrect repair suggestions lead to long debug times and even wrong repair actions, which significantly increase the repair cost and adversely impacts yield.

In this chapter, we introduce a machine learning-based intelligent diagnosis system, which can automatically learn debug knowledge from empirical data and identify the most likely root cause of a new failed board. Using such a diagnosis system eliminates the difficulties involved in traditional knowledge acquisition. Fine-grained fault syndromes extracted from failure logs and the corresponding repair actions are used to train the system. Support vector machines (SVMs) have been used in board-level diagnosis to provide accurate root cause isolation. An SVM-based diagnosis system can be rapidly trained and is scalable to large datasets. However, the SVM method used in prior work [1] was simplistic, relying on an arbitrarily chosen kernel function, and it was not adaptive to the availability of new data or test cases. We propose a diagnosis system based on multi-kernel support vector machines (MK-SVMs) and incremental learning, which are used to tune the diagnosis system in an automatic manner. The MK-SVM method leverages a linear combination of single kernels to achieve accurate faulty component classification based on the errors observed. The MK-SVMs thus generated can also be updated based on incremental learning, which allows the diagnosis system to quickly adapt to new error observations and provide even more accurate fault diagnosis.

The remainder of this chapter is organized as follows. Section 2.1 reviews the background and prior work. Section 2.2 reviews basic concepts in support vector machines. Section 2.3 introduces multi-kernel-based SVMs, and describes how MK-SVMs can be extended for incremental learning, namely iMK-SVMs. Section 2.4 presents experimental results on diagnosis accuracy and training time for two industry

F. Ye et al., *Knowledge-Driven Board-Level Functional Fault Diagnosis*,
DOI 10.1007/978-3-319-40210-9_2

boards and for synthetic data. These results are compared to diagnosis using single-kernel SVMs [1] and ANNs [2]. In addition, experimental results are presented for the diagnosis accuracy achieved using incremental learning. The high diagnosis accuracy, rapid training, and short diagnosis time highlight the benefits of the iMK-SVM-based reasoning system. Section 2.5 concludes the chapter.

2.1 Background and Chapter Highlights

Field data and experience reports from repair technicians highlight many problems with diagnostic software currently in use, especially for functional tests that involve actual data in a real application. The diagnostic resolution offered by today's tools is limited to ASICs on the board. No repair guidance is provided for memory devices or passive components on the board. Diagnostic resolution is also poor in practice, multiple repair candidates are often listed and these candidates are not prioritized. Technicians are forced to run debug programs repeatedly and carry out physical probing in many places to identify the root cause of failures, a practice that significantly increases the debug and repair time. Based on past repair records, we have found the debug time for the functional test considered here to be as high as several weeks. The correctness of diagnosis, i.e., the probability of the actual failing component included in the list of suspects, is unacceptably low, and the root cause is seldom exclusively pinpointed.

In order to overcome the difficulties described above and provide accurate diagnostic results, we investigate intelligent diagnosis based on machine-learning algorithms. Machine learning, a branch of artificial intelligence, is focused on automatic learning from empirical data and making intelligent decisions. The debug knowledge can be automatically learned from history records (logs) using these techniques, e.g., artificial neural networks (ANNs) [2]. In ANN learning, we are given a set of training cases, which typically contain a set of error observations, referred to as syndromes. An ANN aims to automatically generate both the edge weights in the network and a transfer function that allow root-cause identification to be made on the basis of the syndromes. Due to its wide acceptance in the machine learning community and ease of interpretation, ANNs have been used for fault diagnosis [2, 3]. However, ANN-based methods suffer from the inherent theoretical limitations of ANNs that tend to limit their accuracy [2]. Moreover, ANNs require large datasets for training, and large volumes of relevant data are not always available.

Recently, success in board-level functional fault diagnosis has also been reported using SVMs, which constitute a more advanced class of machine-learning techniques [1]. Even though SVMs were shown to be more effective than ANNs for a complex board in high-volume production [1], the increase in success ratio (diagnosis accuracy) was marginal. Moreover, the SVM method used in [1] was simplistic, relying on an arbitrarily chosen kernel function, and it was not adaptive to the availability of new data or test cases.

In this chapter, we propose an adaptive, accurate, and efficient diagnostic system based on SVMs. An advantage of using machine learning is that it avoids the difficulties associated with knowledge acquisition and rule-base development required for expert systems [1, 2, 4]. Without the need to understand the complex functionality of boards, diagnostic systems based on machine learning are able to automatically derive and exploit knowledge from repair logs of previously documented cases. The proposed approach overcomes the limitations of single-kernel SVMs used in [1] by exploiting multi-kernel SVMs and incremental learning (iMK-SVM) to reduce complexity, achieve significantly higher diagnosis accuracy, and perform reasoning adaptively in realtime as new data becomes available. The kernel function in this approach is defined as a linear combination of different kernels. The proposed iMK-SVM-based diagnostic system is generic. Given a set of fault syndromes and the corresponding faulty components, the system can be rapidly trained and then used for fault diagnosis across different products. Results are presented for two industry boards, which are currently in production, and for which fail data has been gathered and used for training and evaluation.

2.2 Diagnosis Using Support Vector Machines

A SVM is a supervised machine learning algorithm proposed by Vapnik in 1995 [5]. It has a number of theoretical and computational merits, for example, the simple geometrical interpretation of the margin, uniqueness of the solution, statistical robustness of the loss function, modularity of the kernel function, and overfitting control through the choice of a single regularization parameter. A brief introduction to SVMs is presented below.

2.2.1 Support Vector Machines

The goal of SVMs is to define an optimal separating hyperplane (OSH) to separate two classes. The vectors from the same class fall on the same side of the optimal separating hyperplane, and the distance from the closest vectors to the optimal separating hyperplane is the maximum among all the separating hyperplanes. An important and unique feature of this approach is that the solution is only based on those vectors that are the closest to the OSH, calculated in the following way. Let $(\boldsymbol{x}_i, y_i)$, $i = 1, 2, \ldots, n$ be a set of training examples, and $\boldsymbol{x}_i \in \boldsymbol{R}^d$, $y_i \in \{-1, +1\}$. Figure 2.1 illustrates a two-class SVM model. The vector $\boldsymbol{x}_i$ is considered as input, and d is the dimensionality of the input vectors. Each input vector belongs to one of the two classes. One is labeled by $y = +1$; the other is labeled by $y = -1$. If the set can be linearly separated, there must be a hyperplane satisfying Formula (2.1):

$$f(\boldsymbol{x}) = sgn(\boldsymbol{\omega}^T \boldsymbol{x} + b), \tag{2.1}$$

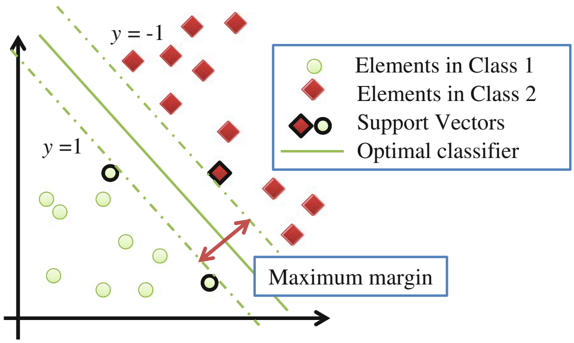


Fig. 2.1 Illustration of a 2-class support vector machine model

where *sgn* refers to the sign of $(\boldsymbol{\omega}^T \boldsymbol{x}+b)$, $\boldsymbol{\omega}$ is a d-dimensional vector, and b is a scalar. Those vectors $\boldsymbol{x}_i$ for which $f(\boldsymbol{x}_i)$ is positive are placed in one class, while vectors $\boldsymbol{x}_i$ for which $f(\boldsymbol{x}_i)$ is negative are placed in another class. Based on [5], we define *margin* as twice the distance from the classifier to the closest data vector, namely the *support vector*. The margin is a measure of the ability to generate a classifier. The larger the margin is, the better is the generation of the classifier. SVMs maximize the margin between two classes.

Since the margin width equals $\frac{2}{\sqrt{\boldsymbol{\omega}^T \boldsymbol{\omega}}}$, the maximum-margin solution is found by solving the following minimization problem:

$$\text{Minimize} \quad W = \frac{1}{2}\|\boldsymbol{\omega}\|^p + C\sum_i^S \xi_i \tag{2.2}$$

Subject to

$$\|y_i(\boldsymbol{\omega} \cdot \boldsymbol{x}_i + b)\| \leq 1 - \xi_i, \ \forall i \tag{2.3}$$

$$\xi_i > 0, \ \forall i \tag{2.4}$$

where slack variable ξ_i is introduced to measure the degree of misclassification of data $\boldsymbol{x}_i$ and C is the error penalty. We can tune C to adjust the trained SVM model to be either overfitting or underfitting. The parameter p is used for regularization of the weights in the SVM model. Most SVM solvers use standard regularization, i.e., $p = 2$. Hence we assume $p = 2$ in our work.

In order to solve the constrained optimization problem described in (2.2), a set of Lagrange multipliers α_i, where $\alpha_i \geq 0$, is used. Each multiplier α_i corresponds to a constraint in (2.3) on the support vectors. The optimization problem from (2.2) can now be expressed in its dual form

$$\text{Minimize} \quad W_1 = \frac{1}{2}\sum_{i,j=1}^S \alpha_i Q_{ij}\alpha_j - \sum_{i=1}^S \alpha_i + b\sum_{i=1}^S y_i\alpha_i \tag{2.5}$$

where $Q_{ij} = y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and K is the kernel function described in the next section. Additional mathematical details are omitted here but they can be found in [5]. The weights and offsets are as follows:

$$\boldsymbol{\omega} = \sum_{i=1}^{S} \alpha_i y_i \boldsymbol{x}_i \tag{2.6}$$

$$b = \frac{1}{N_S} \sum_{i=1}^{S} (y_i - \boldsymbol{\omega} \cdot \boldsymbol{x}_i) \tag{2.7}$$

Originally, SVMs were designed for linear binary classification problems. In practice, classification problems are not limited by two classes. In board-level fault diagnosis, the number of root cause candidates (classes) is typically in the range of a few hundreds. In [5], a modified design of SVMs was proposed in order to incorporate multiclass learning. Besides this, an alternative approach for handling a large number of classes is to combine multiple binary SVMs. "One against one" provides pairwise comparisons between classes. "One against all" compares a given class with all the other classes. According to a comparison study in [5], the accuracies of these methods are almost the same. Therefore, we choose the "one against all" in our problem, which has the lowest computation complexity.

2.2.1.1 Demonstration of SVM-Based Diagnosis System

To illustrate the SVM optimization procedure, consider the same hypothetical demonstration board with six cases in Sect. 2.2. We build an SVM model to identify faults for new cases. Let x_1, x_2, and x_3 be three syndromes. If the syndrome manifests itself, we record it as 1, and 0 otherwise. The presentation of fault class is different from that for ANNs training in Sect. 2.2. The board has two candidate root causes A and B, and we encode them as $y = -1$ and $y = 1$, respectively. In a real scenario, fault syndromes vary across products and tests. Here, we merge the syndromes and the known root causes into one matrix $\mathscr{A}' = [\mathscr{B}'|\mathscr{C}']$, where the left ($\mathscr{B}'$) side refers to syndromes, while the right side ($\mathscr{C}'$) refers to the corresponding fault classes. This matrix represents the training information for the SVM.

$$\mathscr{A}' = \left[\mathscr{B}'|\mathscr{C}'\right] = \left[\begin{array}{ccc:c} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 \end{array}\right] \tag{2.8}$$

We obtain the Lagrange multipliers $\alpha_1 = 2.00e^{-7}$, $\alpha_2 = 1.99$, $\alpha_3 = 1.99$, and $\alpha_4 = 1.99e^{-7}$ by solving the optimization (cost) function from (2.5). We then get

$\boldsymbol{\omega}_1 = 1.99$, $\boldsymbol{\omega}_2 = 0$, $\boldsymbol{\omega}_2 = 0$ and $b = -1.00$ by solving Eqs. (2.6) and (2.7). Therefore, the classifier for determining the root cause for the given set is generated as follows:

$$f(\boldsymbol{x}) = sgn(1.99 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 - 1.00) \tag{2.9}$$

Next, suppose a new failing board is received and it has the syndrome [1 1 0], which corresponds to the first row (case) of $\mathscr{A}'$ in Eq. (2.8). The function y is evaluated using Eq. (2.2), and since $sgn(1.99 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 - 1.00)$ is positive, $y = 1$. Thus the root cause for this failing board is determined to be A. Suppose a second new failing board with syndrome [0 1 0] in received. In this case, the decision function evaluates to $y = -1$, hence we determine B to be the root cause in this case. For boards with the root cause of class A (B), we can replace the corresponding component A (B).

2.2.2 SVM Diagnosis Flow

The SVM-based diagnosis flow consists of four steps that are described in Fig. 2.2. Generally speaking, a set of training data (fault syndromes and corresponding repair actions) is first prepared, which is derived from the repair history in the manufacturing database. Then SVMs determine the OSH based on the training data. After the OSH is determined, the SVMs-based diagnostic system is ready to diagnose new cases.

Step 1: The data preparation step also follows the description in Sect. 2.1. The extracted syndromes and replaced components are used as inputs and outputs for the training of SVMs.

Step 2: Proper kernel function and penalty parameter are chosen to determine the SVMs training. The choice of these two parameters affects the performance of SVMs.

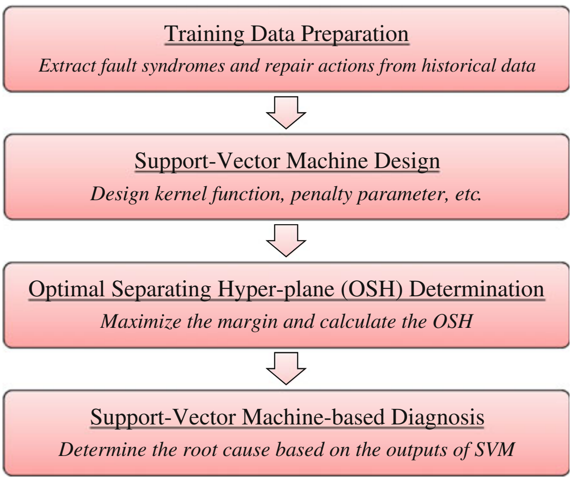


Fig. 2.2 The diagnosis flow using SVMs [1]

However, there are no generic rules to select the best kernel and other parameters of SVMs for a specific problem. In this work, we determine the best design of SVMs in a heuristic way. According to extensive experimental results, we find that the SVMs with a linear kernel and a relatively large penalty parameter provide the highest diagnostic accuracy in the board-level fault diagnosis.

Step 3: The determination of the OSH can be considered as the training of SVMs. The OSH is determined by solving the quadratic optimization problem described in Eqs. (2.7) and (2.6). The values of $\boldsymbol{w}$ and b can be calculated using MATLAB. An open-source SVM toolbox is provided in [6]. The $\boldsymbol{w}$ and b values are determined after training.

Step 4: Given a new input vector, we can calculate the output of the SVM using the decision function in Eq. (2.1). In the diagnosis step, we rank the output of all the SVMs, and select the component represented by the SVM with the largest output as the root cause.

2.3 Multi-kernel Support Vector Machines and Incremental Learning

Classical SVMs are efficient for linear classification, as discussed in Sect. 2.2. However, in many practical scenarios, including fault diagnosis, classical SVMs fail to find an optimal linear classifier for separating classes. Therefore, in such scenarios, SVMs must be extended to handle nonlinear classification problems. One solution is to transform the problem to a higher dimensional feature space through a nonlinear mapping, also known as *kernel*, and the classifier is constructed in the new feature space. The advantage of this transformation is that it is not necessary to explicitly implement the transformation and to determine the separating hyperplane in a higher dimensional feature space.

2.3.1 Multi-kernel Support Vector Machines

2.3.1.1 Kernel

In kernel-based transformation methods, the data representation is implicitly chosen through a *kernel* $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are both input vectors in the lower dimensional feature space. Figure 2.3 illustrates the transformation from a lower dimension feature space to a high-dimension feature space. The optimization problem from (2.5) can now be expressed as:

$$\text{Minimize} \quad W_2 = \frac{1}{2}\sum_{i,j=1}^{S} \alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{S} \alpha_i + b\sum_{i=1}^{S} y_i\alpha_i \tag{2.10}$$

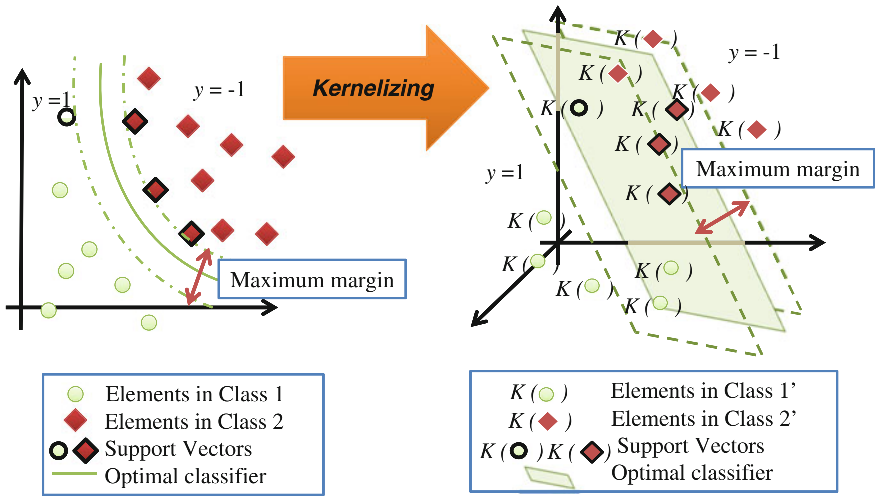


Fig. 2.3 Illustration of kernelized support vector machine model

with similar constraint functions as those for nonkernel SVMs.

$$b = \frac{1}{N_S} \sum_{i=1}^{S} \left(y_i - \sum_{j=1}^{S} \alpha_j y_j \cdot K(\boldsymbol{x}_i, \boldsymbol{x}_j) \right) \tag{2.11}$$

The decision function (2.1) is now expressed as

$$f(\boldsymbol{x}) = sgn\left(\sum_{i=1}^{S} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \right) \tag{2.12}$$

The choice of kernel function is crucial for the success of an SVM-based model. There are several widely used kernel functions [5]:

- Homogeneous Polynomial Kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)^d$, where $d \geq 1$. The linear kernel ($d = 1$) is regarded as a kernelized representation of linear SVMs.
- Polynomial kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^d$, where $d \geq 1$.
- Gaussian kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}$, where $\gamma = \frac{1}{2\sigma^2}$ and σ can be interpreted as the standard deviation of a Gaussian distribution.
- Exponential kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|}$.

As an illustration, we apply kernelization methods to the example described in Sect. 2.2. Suppose that we choose a polynomial kernel with degree 2 and $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^2$. Given a new failing board under test with syndrome [1 1 1], the kernelized presentation of the syndrome is [0.91 0.64 0.64] and $f(\boldsymbol{x})$ is positive. Therefore, we can classify it as being in class A, i.e., the root cause is A. Suppose another new failing board is received with syndrome [0 1 0]; $f(\boldsymbol{x})$ is now negative. Hence, this board is classified as being in class B, i.e., the root cause of failure for this board is B.

2.3.1.2 Multi-kernel Support Vector Machines

Recent applications in bioinformatics have shown that using multiple kernels instead of a single one can lead to better classification [7, 8]. The key idea here is to represent the kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as a linear combination of M basis kernels:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{M} \mu_k K_k(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{2.13}$$

where $\mu_k \geq 0$ and $\sum_{k=1}^{M} \mu_k = 1$. Each basis kernel K_k can be one of the kernel types listed in Sect. 2.3. Thus the optimization problem can now be stated as

$$\begin{aligned} \text{Minimize} \quad W_3 = \frac{1}{2} \sum_{i,j=1}^{S} \alpha_i \alpha_j y_i y_j k \sum_{k=1}^{M} \mu_k K_k(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ - \sum_{i=1}^{S} \alpha_i + b \sum_{i=1}^{S} y_i \alpha_i, \end{aligned} \tag{2.14}$$

The optimization problem of Eq. (2.14) is solved using a reduced gradient method, the details of which are described in [9]. The training mechanism of a multi-kernel SVM-based diagnosis system is illustrated in Fig. 2.4.

In previous work [1], the SVM-based diagnosis system leverages single kernels in a heuristic manner. However, due to the correlation between syndromes, we cannot arbitrarily determine a single kernel for each diagnosis system. A diagnosis system requires an adaptive kernel in order to achieve higher prediction accuracy. Such adaptation can extend from a single kernel to multiple kernels. Without knowledge of the exact kernel to be used, the weights of different kernels can be appropriately configured to fit the training data and generate better prediction for the diagnosis system.

2.3.2 *Incremental Learning*

Incremental learning can not only solve the SVM training problem for large-scale data sets, but it can also facilitate online learning for SVMs as new data for failing boards and the corresponding repair outcomes become available. The use of multi-kernel SVMs increases computational complexity and diagnosis solutions take longer to converge. This problem can be tackled using incremental learning. By distributing

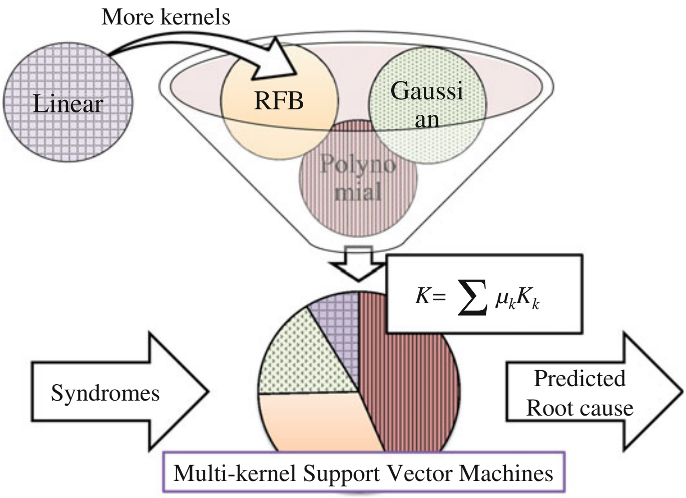


Fig. 2.4 Illustration of training a multi-kernel SVM

the computational workload among different epochs during training, computational complexity can be reduced in terms of both computing time and memory allocation. Since SVM models can be determined on the basis of support vectors only, and the number of support vectors is typically very small compared to the number of training examples, SVMs can benefit from incremental learning through the compression of data from previous batches in the form of their support vectors. This approach to incremental learning with SVMs has been investigated in [10], where it has been shown that incrementally trained SVMs are as effective as their nonincrementally trained equivalents. Incremental SVMs can be described as the following optimization problem as an extension to (2.2):

$$\text{Minimize} \quad W^{\diamond} = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C\left(L\sum_{i\in S^*}\xi_i + \sum_{i\in S}\xi_i'\right) \tag{2.15}$$

with the same constraints (2.3) and (2.4). The parameter S^* denotes the set of existing support vectors extracted from the previous SVM models, and S is the new training set. As an optimization knob, the use of existing support vectors can be penalized by L to model the fact that an erroneous decision made on the basis of previous support vectors is more costly than an error on a new example based on the current data. Incremental learning can be made more effective by combining it with multiple kernels, an approach that we refer to as iMLK-SVMs. A flowchart for this procedure is shown in Fig. 2.5.

As an illustration, consider an existing diagnosis system given by Eq. (2.1) and based on the input cases (boards) represented by the matrix $\mathscr{A}$ in Eq. (2.8). The support vectors in the existing diagnosis system can be extracted as shown in $\mathscr{A}'$.

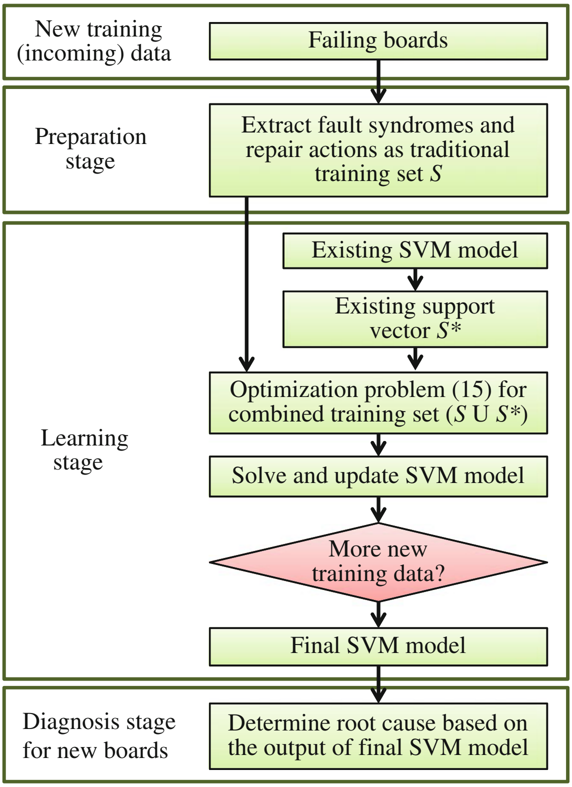


Fig. 2.5 Illustration of the proposed iMK-SVM approach for fault diagnosis

Suppose that we are given four new training boards (cases) indicated by the matrix $\mathscr{N}$ in Eq. (2.3.2) for incremental learning. The classifier (2.1) gives inconsistent root-cause identification—root cause A with the existing data but B with the new data. Therefore, we form a new data set $\mathscr{D} = [\mathscr{A}'; \mathscr{N}]$ as shown in Eq. (2.3.2).

$$\mathscr{A}' = \left[\begin{array}{ccc:c} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -1 \end{array}\right]$$

$$\mathscr{N} = \left[\begin{array}{ccc:c} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \end{array}\right]$$

$$\mathscr{D} = \left[\begin{array}{c} \mathscr{A}' \\ \hdashline \mathscr{N} \end{array} \right] = \left[\begin{array}{ccc:c} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -1 \\ \hdashline 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \end{array} \right]$$

We obtain a unique solution to the new optimization problem from (2.15). We then get $\boldsymbol{\omega}_1 = 0$, $\boldsymbol{\omega}_1 = 0$, $\boldsymbol{\omega}_1 = -2.00$ and $b = 1.00$. The classifier is now updated from (2.1) to be:

$$f(\boldsymbol{x}) = sgn(0 \cdot x_1 + 0 \cdot x_2 - 2.00 \cdot x_3 + 1.00) \tag{2.16}$$

Next, we suppose that a new failing board is received and has the syndrome [1 1 1], which corresponds to the third row (case) of $\mathscr{N}$. The function $f(\boldsymbol{x})$ is now evaluated using Eq. (2.16) and its value is negative, rather than the positive prediction based on the previous decision function of (2.1). Therefore, we can conclude that the failing board is in Class B, i.e., the root cause is B. Suppose another failing board is received with syndrome [1 1 0], which is the first test case from Sect. 2.2. In this case, the new decision function still evaluates $f(\boldsymbol{x})$ to be positive; hence, we determine A to be the root cause for this board.

2.4 Results

Experiments were performed on two industrial boards that are currently in high-volume production. Relevant information about the boards is provided in Table 2.1. For training, a total of 1613 repaired boards are collected from the contract manufacturer's database for Board 1. A total of 546 fault syndromes are extracted from failure logs. The number of faulty components identified in the database for repair action is 153. For Board 2, a total of 1351 repaired boards are analyzed as training data. A total of 375 fault syndromes are extracted from failure logs. The number of faulty components for repair action is 116.

Table 2.1 Information about the industrial boards used for classification and the log data available

	Board 1	Board 2
Number of syndromes	546	375
Number of repair candidates (components)	153	116
Number of boards	1613	1351

The SVM algorithms are implemented using the MATLAB 2010b toolbox. Multi-kernel SVMs are implemented using SimpleMKL [9]. Incremental learning is implemented using McpIncSVMs [11] and the SVM-KM toolbox [6]. As a comparison, the ANN method described in [2] has also been implemented using the Neural Network toolbox in MATLAB 2010b [12]. Experiments were run on a 64-bit Windows system with 12 GB of RAM and quadcore Intel i7 processors running at 2.67 GHz. Diagnosis results were obtained for different designs of the SVMs, e.g., for various kernel functions. Moreover, diagnosis results were obtained to highlight the comparison between traditional artificial neural networks and the proposed multi-kernel SVMs. Incremental SVM-based diagnosis results were next compared with nonincremental SVM-based methods. Diagnosis results show that the training time is reduced significantly if we implement incremental learning in linear-kernel SVMs. Furthermore, iMK-SVM-based diagnosis results show high classification rates but low training time in each epoch, thereby providing a practical method for designing an adaptive diagnostic system.

In order to assess the performance of the classifier and its ability to accurately predict the root cause of a failure on a new board, we use a *cross-validation* method to partition the training set into k groups, namely k-folder cross-validation [13]. Each group is regarded as a test case while all of the other cases are fed for training. In our work, we assess our model by using a special type of cross-validation method, namely *leave-one-out* (LOO), where the number of partitions k is the same as the total number of cases. For example, for Board 2, each board instance is iteratively selected to be the test case. Classification models are based on the remaining 155 training cases. In LOO estimation, the total number of cases in the testing set is same as the total number of available successfully repaired boards.

To ensure real-time diagnosis and repair, we assume that we are allowed at most three attempts to replace the potential failing components. Success ratio (SR) is the ratio of the number of correctly diagnosed cases to the total number of cases in the testing set. We define SR_1 as the success ratio corresponding to the case that the board is deemed to have been successfully repaired only when the actual faulty component is identified and placed at the top of the list of candidates. We also define SR_2 (SR_3) as the success ratio corresponding to the case that a board is deemed to have been successfully repaired if the actual faulty component is in the first two (three) positions in the list of candidates. In the last column in Table 2.2a, we can see that SR_1 is 80.2 %. If three attempts are allowed in the repair process, 91.9 % of the boards can be successfully repaired. These results are a significant improvement over other approaches reported recently in the literature [1, 2]. The SR values for Board 2 are lower (Table 2.2b). Nevertheless, tangible improvement is obtained over other methods, and the diagnosis accuracy is higher than that for methods currently used in production. The training data and evaluation methods will also continue to be used in the following chapters.

Table 2.2 Diagnosis results using ANN, SVMs with different kernel functions, and multi-kernel SVMs

	ANN	SVM methods							
		Linear kernel	Polynomial kernel			Gaussian kernel			Multi-kernel
			$d = 2$	$d = 3$	$d = 4$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 5$	
(a) Board 1									
SR_1	67.9 %	73.2 %	74.4 %	72.3 %	74.9 %	62.6 %	65.1 %	62.6 %	80.2 %
SR_2	78.1 %	80.4 %	82.2 %	81.2 %	82.7 %	74.0 %	76.3 %	74.5 %	85.4 %
SR_3	84.4 %	88.2 %	91.9 %	90.4 %	91.9 %	79.2 %	82.5 %	79.3 %	91.9 %
Training time (s)	71.2	43.2	45.2	41.1	42.0	49.9	50.3	50.1	5963
(b) Board 2									
SR_1	57.9 %	66.3 %	63.2 %	63.2 %	66.3 %	66.3 %	60.3 %	55.4 %	71.4 %
SR_2	70.3 %	74.3 %	70.1 %	70.1 %	75.5 %	74.3 %	67.8 %	67.8 %	77.8 %
SR_3	75.1 %	84.1 %	79.5 %	78.5 %	82.1 %	83.5 %	72.6 %	70.2 %	83.7 %
Training time (s)	60.2	23.6	25.9	21.6	22.5	29.1	35.4	36.3	3891

2.4.1 Evaluation of MK-SVM-Based Diagnosis System

We use a combination of seven kernels in the multi-kernel machine, including linear kernel, Gaussian kernel (σ values of 0.5, 1, and 5), and polynomial kernel (degree $d = 2, 3, 4$). Diagnosis results are shown in Table 2.2a for Board 1 and Table 2.2b for Board 2. For Board 1, the SR_1 for the multi-kernel SVM is 6–18 % higher than for the single-kernel SVMs and the ANNs. When the SR_3 is considered, the performance of multi-kernel SVMs is similar to that for the single-kernel SVMs. For Board 2, similar improvement of diagnosis accuracy by using multi-kernel SVM is obtained in Table 2.2b. The use of multi-kernel SVM technology leads to a considerable improvement in diagnosis accuracy, but the training time of the multi-kernel SVMs is higher compared to single-kernel SVMs and the ANN. For example, training MK-SVM for Board 1 requires up to an hour as compared to only tens of seconds using single-kernel SVMs and ANNs. Since the training time depends on the number of root causes, number of syndromes, number of cases, and number of iterations required for convergence as described in Sect. 2.3, the training time of multi-kernel SVMs increases quadratically with the board complexity and the number of failing boards that are returned for repair.

2.4.2 *Evaluation of Incremental SVM-Based Diagnosis System*

We implemented linear SVM training on Board 1 and Board 2. The results of incremental learning for linear-kernel SVMs are shown in Figs. 2.6 and 2.8 for Board 1, and in Figs. 2.7 and 2.9 for Board 2. For example, we have a total of 813 training cases for Board 1 as described in Table 2.1. Initially, 800 training cases are randomly selected to build a base SVM model. In the second epoch, 100 more new training cases are randomly selected from the remaining pool. The existing SVM model is updated by using incremental learning to append these new training cases. A total of 100 more new training cases are added into SVM models in the next epoch and so forth. We also construct a nonincremental learning SVM from scratch in each epoch with the same number of training cases as in the corresponding incremental learning SVMs. Figure 2.6 shows that the training time for nonincremental learning SVMs increases linearly with the number of training cases, but the training time for incremental learning remains nearly constant in each epoch, even though the number of training cases increases. In the last epoch when 100 more training cases are appended to the existing SVM with 1500 training cases, the training time of SVMs using incremental learning is 8.27 s compared to 43.2 s using nonincremental learning. This quantifiable reduction in training time using incremental learning can be a significant benefit if thousands of failing boards are returned for repair in high-volume manufacturing. Incremental learning also reduces computational complexity and memory required for training, as described in Sect. 2.3. The use of incremental learning can help reduce the computational complexity of SVMs as described in Sect. 2.3.

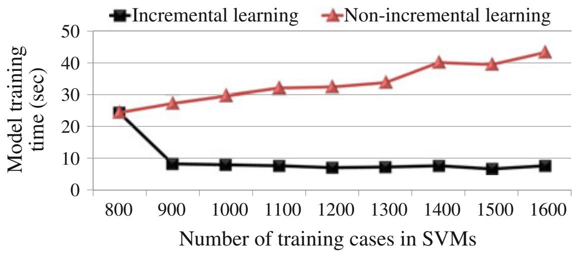


Fig. 2.6 Comparison of training time in each epoch between incremental and nonincremental linear-kernel SVMs for Board 1

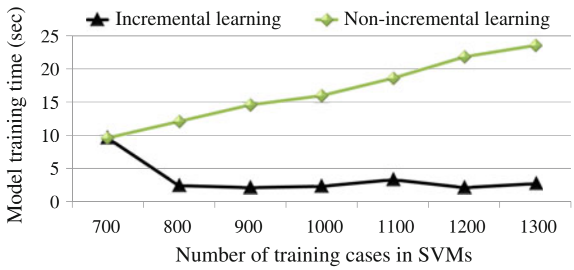


Fig. 2.7 Comparison of training time in each epoch between incremental and nonincremental linear-kernel SVMs for Board 2

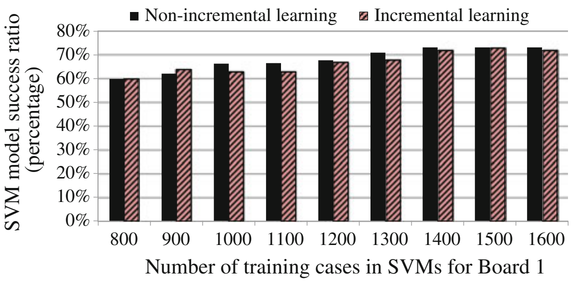


Fig. 2.8 Comparison of SR_1 between incremental and nonincremental linear-kernel SVMs for Board 1

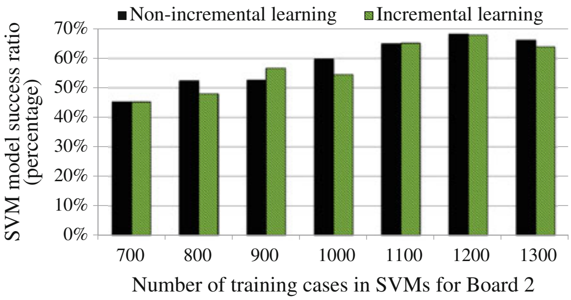


Fig. 2.9 Comparison of SR_1 between incremental and nonincremental linear-kernel SVMs for Board 2

The comparisons of SR_1 between incremental learning and nonincremental learning are shown in Figs. 2.8 and 2.9. We found that there is little difference in SR_1 between incremental learning and nonincremental learning, thus nonincremental learning can be replaced by incremental learning with the similar success ratios, but the training time is reduced using incremental learning. The observations on SR_2 and SR_3 are the same as that on SR_1. We also found that the SR_1 increases when more failing boards are fed to the SVM for training. The diagnosis success ratio is strongly correlated to the size of the training set [1]. The SR_1 of the linear-kernel SVMs increases from 59.5 % with training cases of 800 to 73.2 % with training cases of 1600 in Fig. 2.8 for Board 1, and from 45.2 % in Table 2.2b to 66.3 % in Fig. 2.9 for Board 2. The increasing trends of success ratios can also be found in SR_2 and SR_3 for both Board 1 and Board 2.

In order to evaluate our results in a practical context, we consider what is typically carried out in a board manufacturing line. Most diagnosis and repair actions still rely on the technician's experience and trial-and-error methods. Current diagnostic software used in the production line from where we obtained the boards considers any component that exhibits error as a fault candidate, and no suggestions are provided regarding which component is more likely to be the root cause. Compared to the SR_1 of the current diagnostic method, the SR_1 for the proposed method is about two times higher, and significantly higher than even the SR_3 of the currently used diagnostic method.[1] Based on the repair suggestions provided by the currently deployed method,

[1]Exact success ratio for the deployed system are not presented here in order to protect company confidential data.

the debug time for this particular functional test is as high as several weeks, which is clearly not feasible in practice. Debug efficiency is therefore expected to be improved considerably with the accurate repair suggestions provided by the proposed method.

2.4.3 Evaluation of Incremental MK-SVM-Based Diagnosis System

Incremental learning can also be applied to MK-SVM training. The success ratio results are shown in Fig. 2.10 for Board 1 and in Fig. 2.11 for Board 2. When we increase the size of the training set, the success ratios of up to three attempts for both Board 1 and Board 2 increase. This observation supports the positive correlation found in [1] between the number of failing boards available for training and the diagnosis accuracy on new boards. The SR_1, SR_2, and SR_3 are 80.2, 85.4, and 91.9 %, respectively, when 1600 training cases are used in the iMK-SVM model for Board 1, and 71.4, 77.8, and 83.7 %, respectively, when 1300 training cases are used in the iMK-SVM model for Board 2.

The training time results are shown in Fig. 2.12 for Board 1 and in Fig. 2.13 for Board 2. The training time of incremental learning MK-SVMs in each epoch is much smaller than that for nonincremental learning MK-SVMs. The training time varies in each epoch for Board 1 due to the size of the training data and iterations needed for convergence, as described in Sect. 2.3. Due to the reduction of support vectors by

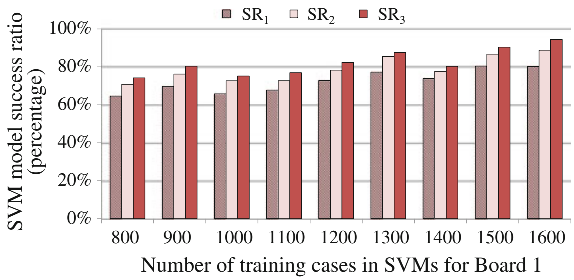


Fig. 2.10 Success ratio of using incremental multi-kernel SVMs for Board 1

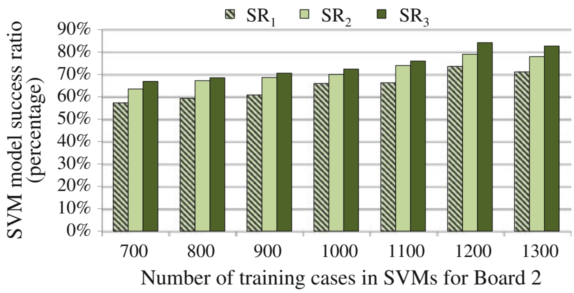


Fig. 2.11 Success ratio of using incremental multi-kernel SVMs for Board 2

Fig. 2.12 Comparison of training time in each epoch between using incremental and nonincremental multi-kernel SVMs for Board 1

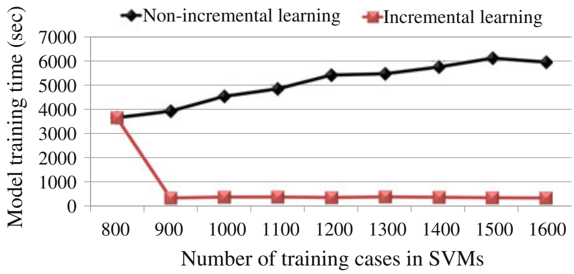
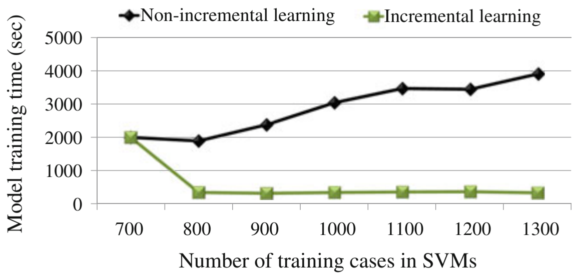


Fig. 2.13 Comparison of training time in each epoch between using incremental and nonincremental multi-kernel SVMs for Board 2



using incremental learning techniques, the effect of iterations on the total training time can also be reduced in iMK-SVMs, as shown in Fig. 2.12.

The weights in iMK-SVMs training change in each epoch. For example, in Fig. 2.14, we use a combination of 13 kernels in a multi-kernel machine, including the linear kernel, Gaussian kernel (σ values of 0.5, 1, 2, 5, 10, 15, 20, and 50), polynomial kernel (degree $d = 2$, 3, and 4), and homogenous polynomial kernel (degree $d = 2$). Only four kernels out of the total set of 14 kernels contribute to the multi-kernel machine; these are the linear kernel, Gaussian kernel with $\sigma = 2$, homogenous polynomial kernel (degree $d = 2$), and polynomial kernel (degree $d = 2$). The weights of the remaining ten kernels are reduced to 0 in the optimized solution. In Fig. 2.14, the weight of the Gaussian kernel with $\sigma = 2$ is 12 % and the weight of homogeneous polynomial kernel with degree =2 is 22 % in the first epoch. When more cases are fed for training, the weights of these two kernels are gradually reduced to 0. In the last epoch, only two kernels are left in the multi-kernel machine. And the dominating kernel is the polynomial kernel with degree =2 (61 %). Furthermore, kernel distribution is different for different board types. For Board 2, the linear kernel, the Gaussian kernel with $\sigma = 5$, polynomial kernel with degree =3, and homogeneous polynomial kernel with degree =2 equally contribute to the multi-kernel in the first epoch. After three epochs, the weight of the homogeneous polynomial kernel is reduced to 0. In the final epoch, when 1300 cases are used for training, the weight of linear kernel is 65 % and dominates the diagnosis system (Fig. 2.15).

Since the optimal classifier solutions for different boards lead to different combinations of kernels, we cannot arbitrarily determine a best single kernel for all

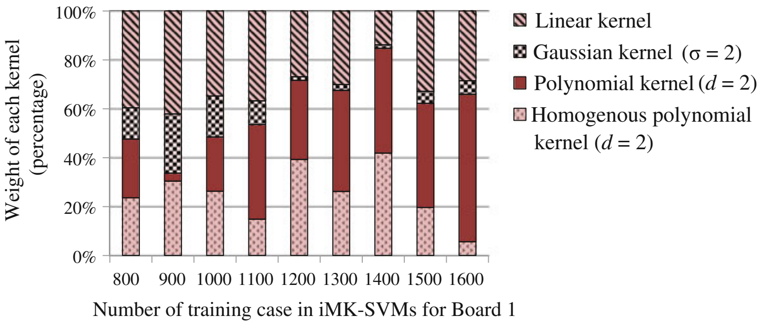


Fig. 2.14 Illustration of the change in kernel weights in incremental multi-kernel SVMs for Board 1

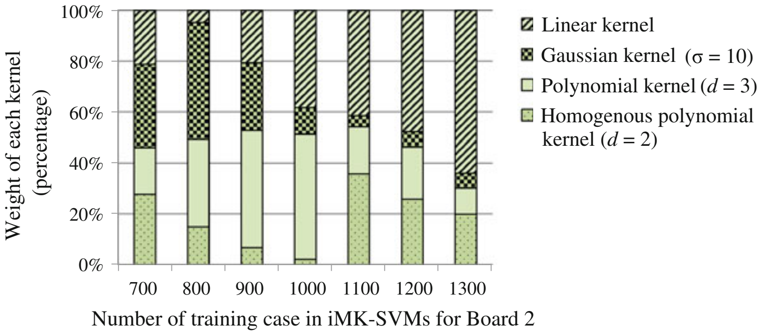


Fig. 2.15 Illustration of the change in kernel weights in incremental multi-kernel SVMs for Board 2

the boards. The use of iMK-SVMs can adaptively choose the most suitable kernel portfolio for different boards.

2.5 Chapter Summary

This chapter has presented a smart diagnosis system based on multi-kernel support vector machines and incremental learning to locate the root cause of functional failures on modern circuit boards. The proposed multi-kernel SVMs method can generate an optimal kernel portfolio to achieve high diagnosis accuracy for board-level functional tests. The use of incremental learning allows the system to adaptively tune the kernel portfolio to achieve high diagnosis accuracy. System training time can also be reduced significantly using incremental learning. Two industrial boards, which are currently in high-volume production, and additional synthetic boards have been used to validate the effectiveness of the diagnosis method. Compared to baseline ANN and several single-kernel SVMs, multi-kernel SVMs show a considerable improvement in diagnostic accuracy based on functional patterns for a real application.

References

1. Zhang Z, Gu X, Xie Y, Wang Z, Chakrabarty K (2012) Diagnostic system based on support-vector machines for board-level functional diagnosis. In: Proceedings IEEE European test symposium (ETS), pp 1–6
2. Zhang Z, Chakrabarty K, Wang Z, Wang Z, Gu X (2011) Smart diagnosis: efficient board-level diagnosis and repair using artificial neural networks. In: Proceedings IEEE international test conference (ITC), pp 1–10
3. O'Farrill C, Moakil-Chbany M, Eklow B (2005) Optimized reasoning-based diagnosis for non-random, board-level, production defects. In: Proceedings IEEE international test conference (ITC), pp 173–179
4. Zhang Z, Wang Z, Gu X, Chakrabarty K (2010) Board-level fault diagnosis using Bayesian inference. In: Proceedings IEEE VLSI test symposium (VTS), pp 1–6
5. Vapnik V (1995) The nature of statistical learning theory. Springer, Berlin
6. Rakotomamonjy A, Canu S (2008) SVM and Kernel Methods Matlab Toolbox. http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html
7. Lanckriet G, De Bie T, Cristianini N, Jordan M, Noble W (2004) A statistical framework for genomic data fusion. Bioinformatics 20:2626–2635
8. Varma M, Babu BR (2009) More generality in efficient multiple kernel learning. In: Proceedings of ACM international conference on machine learning (ICML), pp 1065–1072
9. Rakotomamonjy A, Grandvalet Y, Bach F, Canu S (2008) SimpleMKL. J Mach Learn Res 9:2491–2521
10. Chang T, Liu H, Zhou S (2009) Large scale classification with local diversity AdaBoost SVM algorithm. J Syst Eng Electron 20(6):1344–1350
11. Incremental SVM Learning with multiclass support and probabilistic output (2013) http://www-ti.informatik.uni-tuebingen.de/spueler/mcpIncSVM/
12. Neural Network Toolbox (2012) http://www.mathworks.com/products/neuralnet/
13. McLachlan G, Do K, Ambroise C (2004) Analyzing microarray gene expression data, vol 422. Wiley, Hoboken

http://www.springer.com/978-3-319-40209-3

Knowledge-Driven Board-Level Functional Fault
Diagnosis
Ye, F.; Zhang, Z.; Chakrabarty, K.; Gu, X.
2017, XIII, 147 p. 75 illus., 65 illus. in color., Hardcover
ISBN: 978-3-319-40209-3