Application of Support Vector Machines to Fault Diagnosis and Automated Repair

C. Saunders and **A. Gammerman**
Royal Holloway,
University of London,
Egham, Surrey,
England
{C.Saunders,A.Gammerman}@dcs.rhbnc.ac.uk
Tel: +44 1784 443437
Fax: +44 1784 439786

H. Brown and **G. Donald**
IFR Ltd.
Longacres House
Norton Green Road
Stevenage, Herts, England
{Brown_H_T,Donald_G_W}@ifrinternational.co.uk
Tel: +44 1438 742200
Fax: +44 1438 772038

Abstract

In this paper we consider the benefits of applying modern machine learning techniques to the problem of Fault Diagnosis and Automated Repair. In the modern manufacturing environment, many aspects of the production line are logged automatically by various systems. These records are put to a multitude of uses including assisting stock control, and monitoring and improving the manufacturing process. This approach has lead to the accumulation of a huge amount of high-dimensional data, and does require new methods to handle it. In this paper we ask if the information commonly held by many companies can be used to assist the repair of faulty products on the production line. We examine the possibility of using pattern recognition techniques to determine correct repairs for faults from past production history. The relative merits of this method compared to other approaches (such as model-based reasoning) are also discussed. Finally, we give some preliminary results which indicate that pattern recognition methods such as the highly acclaimed Support Vector machine can be successfully applied in this area.

Introduction

Companies are being driven to reduce the cost of their production and are concerned about getting their product to their customers quickly and efficiently. Electronic Assembly manufacturers are using the information from their test systems as a "window on the goodness of their assembly process". After all, if they could build product correctly 100% of the time they wouldn't need to test it. If the data from the test system is automatically logged to a data base, then any faulty items could be sent to a repair station and the faults associated with that item called up on a Paperless Repair screen. This reduces the amount of paper on the shop floor and offers the operator the opportunity to enter the repairs carried out on the item. This information can then be fed back upstream in the manufacturing process to allow assembly workers to identify which process steps are inducing failures. One such mechanism for logging data from test equipment and presenting information to repair technicians is the IFR i-Base Information Management product which was used to gather the information for this research. In this paper we will examine the role that Support Vector machines (Vapnik 1995) can play in this environment of electronic assembly manufacture. Consider the process of manufacturing a specific product. The product will progress down the line being modified and tested at various stages in its creation. If everything runs smoothly the product will be successfully completed and will not have failed any tests en route. At some stage however, a product may fail a test, and then it is up to a technician to determine what's wrong and repair the fault. This process of diagnosis and repair is very costly, both in time and in the cost of the technician. If the process could be totally automated, or information could be presented to the technician in order for them to repair the fault quickly, that would be of great benefit. We will first define precisely what is meant by fault diagnosis and automated repair in this context. The properties of the problem will also be examined, and the suitability of a "learning by examples" approach compared to other methods such as model-based reasoning will be discussed. Thirdly we will describe some simple statistical methods which have been used to tackle the problem, and SV machines will also briefly be described. Results on a small but representative data set are then presented, which gives empirical evidence to successfulness of this approach.

Fault Diagnosis and Automated Repair

Fault diagnosis is the procedure of discovering the exact nature of why a particular piece of equipment is not operating as expected. Fault diagnosis is a widely researched area and many different approaches have been tried. These include model-based approaches (see e.g. (de Kleer 1990; de Kleer & Williams 1987)), cased-base reasoning, and expert systems among others. There are also links between diagnostic/repair processes and formal methods (Gertz & Lipeck 1995). The main objective of many of these systems is to locate and diagnose the exact fault of the device. In this paper we are interested in a subtly different scenario. We are not so much concerned with the prediction of the exact nature of the fault, but the task of identifying the correct repair action. That is, can we provide helpful information to a technician repairing products on a production line, based on previous faults and repairs. Many of the ideas and discussion presented here is applicable to the manufacture of various products on a production line. Here however, we are specifically going to concentrate on an application which uses the manufacture of electronic products as its base. In order to define the problem more concisely, we will first need to give some problem specific background.

Application specifics

At various stages in the production process an electronic device will be tested in various ways until the product reaches the end of the line where a final test will deem it as working correctly. A typical product will start life as one or more printed circuit boards (PCB's). In the early stages these can often be tested individually and an in-circuit test (ICT) is often carried out. Here the PCB is placed on a "bed of nails" and an automatic tester (ATE) is programmed to check each component on the board individually. It is often the case that if the a board fails an ICT, then the component responsible is easily highlighted (as the bed of nails allows access to each component individually). In some cases however, the failure of one or more components can have a knock-on effect and the fault cannot easily be determined. Later on in the production stage, a product may become more complicated and an ICT may not be possible, as access to components may be restricted (e.g. our single board has now become an assembled set). In this case Functional Tests (FCT's) are carried out. This involves stimulating "input" sections of the board which can be reached, and measuring a corresponding output. Here the correct repair is not as self-evident as it was for an ICT test. The fault will be determined by subtleties in the different outputs from the board, which are harder to combine and reason to a fault. In this paper the main aim is to try and predict the correct repair action when a set of faults occurs (either from an ICT or an FCT), based on the past history of the production line. In this scenario, the technician repairing the device may not need to know the exact nature of the fault. Indeed the only information needed to ensure the device progresses to the next stage of production is to identify the correct repair action and to carry it out. In this paper the problem of identifying the correct repair action is what we term to be fault diagnosis and repair.

Learning from Examples

Why use a technique which learns from examples, rather than one which models the device itself? Modelling is a lengthy process which takes a great deal of effort and time. Modelling is also to a large extent device-specific. For new devices, modifications to an existing model or a completely new one will have to be created. This is therefore only beneficial if the modelling stage is only a small fraction of the total life of the production. A more generic approach which can cope with any device, and could be trained quickly without human interaction, would be more beneficial. The open question remains however as to whether a system with no device specific knowledge can successfully learn from a repair history. Some results using this approach with Neural Networks (Totton & Limb 1991) suggest that it is successful, and we continue in the same spirit here. On many production lines, a vast amount of data is automatically logged by various systems. These are often used by companies to re-order components when stock is low, monitor and improve the manufacturing process, and so on. In the production of electronic devices this is especially evident, as most of the tests on the equipment are performed automatically and the data is often logged to a networked storage medium. In a similar fashion, any repairs carried out are also logged, for stock and performance purposes. Given that this data exists and is frequently being used, it should also be possible to use it to assist the technicians on the production line. A system which can successfully learn from examples, would be able to highlight correct repair actions irrespective of the type of device, with only the overhead of training on the database, and without relying on gaining information from experts.

Data Representation

In a majority of cases, the problem of assigning repair actions can be expressed as a pattern recognition problem. Specific faults on a device can be seen as attributes which describe the object, and the repair action can be interpreted as the class in which it belongs. One possible representation is therefore to have a vector which has the same dimension as the number of possible faults (if known, or as an approximation, the number of different faults observed so far), and for each entry have a pass/fail indicator[1]. As a test will often fail in a small number of ways, this leads to large, but very sparse vectors in the data set. Of course, a device may require more than one repair action, and will then be seen to belong to more than one class. This representation (sparse vectors belonging to multiple classes), is very similar to those used successfully in the problem of text categorisation (see e.g. (Joachims 1997)).

The Test-Fail-Repair Cycle

Before we proceed, one more detail concerning representing the problem as described above has to be discussed. The representation above works well in a majority of cases, those primarily being the situations where a board has one or more failures and these are corrected by a well-carried out repair. Unfortunately, things are not always that simple, and a product may fail a set of tests and be repaired several times until the board either passes or is deemed un-repairable. This process is illustrated in figure 1. There are several reasons why

[1] Obviously the naive pass/fail approach can be extended by including more information, e.g. fail high/low, specific voltage of failed component etc.

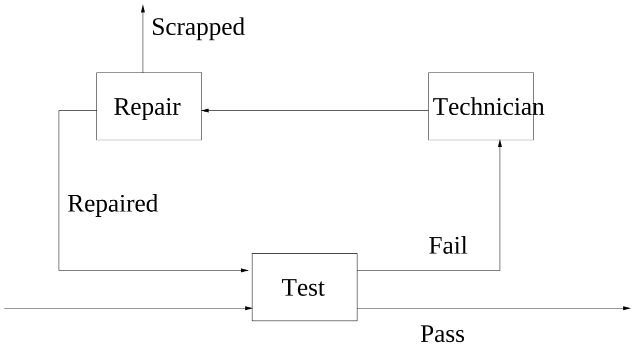

Figure 1: The Test-Fail-Repair Cycle

a certain product will go through the test-fail-repair cycle before moving on to the next stage of production. Principally these are

1. An incorrect repair was made.
2. The repair was carried out badly and did not correct the fault.
3. The fault may have successfully been corrected, however there could be masked faults which are only become apparent after the repair was made.

These situations give rise to problems when converting a log file into a data set containing training vectors, as all of the situations hi-lighted above give rise to multiple entries in the file for a single product. One stratagem is to include only the last entry in the log file (i.e. the set of faults and associated repair, after which the product passed). Another method would be to include several entries. The first entry would include only those faults which the first repair corrected, the second with only those faults not apparent after the second repair, and so on.

Support Vector Machines

Support Vector machines (Vapnik 1995; 1998) have been demonstrated to be an effective tool when applied to a variety of applications in pattern recognition. These include the recognition of handwritten digits (Schölkopf 1997) and 3-D objects (Roobaert 1999), Breast Cancer prognosis (Friess, Cristianini, & Campbell 1998), and engine-knock detection (Rychetsky, Ortmann, & Glesner 1999) Here we will briefly describe the Support Vector algorithm for pattern recognition. Suppose we have a data set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l), \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$. We want to find a hyperplane which separates the data and does so with maximal margin. This can be achieved by minimising

$$\frac{1}{2}||\mathbf{w}||,$$

under the constraints,

$$y_i[(\mathbf{w} \cdot \mathbf{x}_i)] + b \geq 1.$$

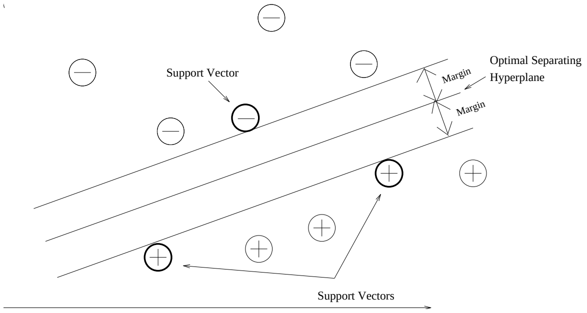

Figure 2: The SV machine solution finds the optimal separating hyperplane between two classes.

If the data are not linearly separable one can introduce slack variables (ξ_i) and instead minimise a modification of the above.

$$\min \frac{1}{2}||\mathbf{w}|| + C \sum_{i=1}^{l} \xi_i,$$

under the constraints,

$$y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 - \xi_i,$$

where C is a constant chosen a-priori. In order to solve this problem, one can introduce Lagrange multipliers and consider the dual problem,

$$\max \sum_{i=1}^{l} \alpha_i - \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j),$$

under the constraints,

$$0 \leq \alpha_i \leq C, \; \sum_{i=1}^{l} \alpha_i y_i = 0.$$

This process has the effect of creating a maximal margin classifier. That is, the two classes presented to the algorithm are separated by a decision hyperplane that achieves the maximum separation of the two classes. This is the so-called optimal separating hyperplane and is shown in Figure 2.

It is possible to move from the restriction of linear hyperplanes by first non-linearly transforming the data into a higher dimensional feature space by using some mapping $\phi : \mathbb{R} \rightarrow \mathcal{F}$. As the optimisation only requires to calculate dot products of these vectors, the mapping function need not be known explicitly. It is sufficient to have a *kernel function* of the form $\mathcal{K}(\mathbf{x}, \mathbf{x}_i) = (\phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i))$ which corresponds to calculating a dot product in the feature space and avoids having to compute the mapping ϕ explicitly. The optimisation problem now is,

$$\max \sum_{i=1}^{l} \alpha_i - \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j),$$

under the constraints,

$$0 \leq \alpha_i \leq C, \; \sum_{i=1}^{l} \alpha_i y_i = 0.$$

This gives rise to the decision rule

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i \mathcal{K}(\mathbf{x}, \mathbf{x}_i) + b \right).$$

Given that Support Vector machines construct a classifier in a high-dimensional space, one might expect that their generalisation ability would be restricted because of the high number of parameters. The key to the generalisation ability of Support Vector machines though lies not in the dimensionality of the feature space, but in the VC-dimension (Vapnik-Chervonenkis dimension, see e.g. (Vapnik 1995)) of the machine. Vapnik has shown that the following inequality holds with probability at least $1-\eta$, and gives a bound for the probability of errors on a testing set:

$$Pr(\mathrm{err}_{\mathrm{testset}}) \leq Pr(\mathrm{err}_{\mathrm{trainset}}) + \phi(\frac{l}{h}, \frac{-\ln \eta}{l}),$$

where h is the VC-dimension of the set of functions, and l is the number of examples in the training set. For the problem of fault diagnosis, the training vectors $\mathbf{x}_i$ will consist of pass/fail information for each test performed on the board, cf. section *data representation*, and the class y will be the repair action carried out. Although the above optimisation procedure only deals with two classes (repair actions), it is easy to generalise this to the multi-class case. Suppose we have n classes, then we can construct n support vector machines, each of which separates one class from all the others. When a new test example is encountered it is classified by each of the SV machines. The machine which classifies it as positive (i.e. a member of that class) with the maximum margin, is deemed to be the correct classification.

Experiments and Results

Experiments were conducted on a small data set (58 examples, 13 attributes, 7 classes), taken from a test board where faults were deliberately introduced so that in all cases the true nature of the fault was known.

Experimental Set-up

One known good test subject was fitted to one of the IFR Automatic Test Equipment (ATE) systems and the test program executed. At the beginning of each run the test program requested an unique board serial number to be entered. This number was incremented at each run. This resulted in several runs yielding passed test results for what appeared to be boards of different serial numbers. One of the components on the board was tampered with such that the test program would fail. The test program was run, the unique serial number noted, and a set of failing results recorded. The i-Base Paperless Repair facility was entered and a description of the injected fault recorded against the particular serial number. The fault was repaired and the test program rerun using the noted serial number. This resulted in the board having a passed status to indicate a successful repair had been carried out. This process was repeated for a number of different faults in addition to using the same fault on a number of different occasions. The test program stored the actual values of passing, as well as failing results, to provide a complete set of information to the diagnostic package. This therefore simulates the actual process carried out on a typical production line.Data was then extracted from the database generated by the i-Base facility, and converted into the representations discussed earlier.

Method	Total Error	Max Error
%-based	13.0%	60%
pattern-matching	6.5%	40%
C4.5	4.0%	20%
SVM	1.0%	20%

Table 1: Total and maximum error rates for the 4 different methods on 20 random splits of the data set

Techniques Used

In order to obtain a measure of the success of the technique, four different methods were applied to the data :

1. A %-based technique which looks at each fault individually and calculates the number of times that each repair action was applied to that fail and was successful. The percentages obtained for each repair action on each fault are then totalled up, and the repair action with the highest percentage is predicted as correct.

2. A simple look-up table which matched fault patterns seen before, and if conflicts arise, it predicts the one which most frequently corrected the problem. If the fault pattern has not been seen before, a repair action is randomly selected.

3. The decision tree algorithm C4.5 (Quinlan 1993), which is known to perform well on pattern recognition problems.

4. Support Vector Machines (described in the previous section).

All four methods used the same data representation, and for consistency they were run on 20 random splits of the data, each time reserving 48 for training and 10 for testing. The results of the experiments are summarised in table 1. The maximum error quoted in the table is the largest error suffered by the algorithm on any single test run. All of the methods achieved 0 error on at least one test run.

Discussion

In this paper we have shown that in principle, pattern recognition techniques can be successfully applied to the problem of fault diagnosis and automated repair. Specifically, we have shown that one of the current leading pattern recognition methods, namely Support Vector machines, are well suited to this task. In many production environments, specifics of device failure and repair are electronically stored and often used for processes such as stock control, or production process planning. In many cases, there will often be sufficient information in the existing database which can be extracted and used in the manner described in this paper. The ability to prompt inexperienced repair technicians to make an effective repair reduces the time an faulty item spends in the repair loop. The ability to access accurate repair information in the form of Pareto graphs allows action to be taken to improve the assembly processes and improve the quality of the product. We would also like to provide additional confidence measures for each prediction given by the algorithm, as this would greatly enhance the usability of a decision support system which used this technique. The support vector approach can easily be extended to provide such values, see (Saunders, Gammerman, & Vovk 1999) for details. Future work will be to incorporate the confidence measures, and to integrate this approach into an online hints facility within a production environment.

References

de Kleer, J., and Williams, B. 1987. Diagnosing multiple faults. *Artificial Intelligence.*

de Kleer, J. 1990. Using crude probability estimates to guide diagnosis. *Artificial Intelligence.*

Friess, T.; Cristianini, N.; and Campbell, C. 1998. The kernel–adatron algorithm: A fast and simple learning procedure for support vector machines. In *Proceedings of the fifteenth International Conference on Machine Learning, ICML '98*, 180–196.

Gertz, M., and Lipeck, U. W. 1995. A diagnostic approach to repairing constraint violations in databases. In *Sixth International Workshop on the Principles on Diagnosis, (DX '95)*, 65–72.

Joachims, T. 1997. Text categorization with support vector machines: Learning with many relevant features. Technical report, Universitat Dortmund.

Quinlan, J. R. 1993. *C4.5 - Programs for Machine Learning.* Morgan Kaufmann.

Roobaert, D. 1999. Improving the generalization of support vector machines: An application to 3d object recognition with cluttered background. In *Proceedings of the Workshop on SV Machines at IJCAI '99*, 29–34.

Rychetsky, R.; Ortmann, S.; and Glesner, M. 1999. Support vector approaches for engine knock detection. In *International Joint Conference on Neural Networks (IJCNN 99).*

Saunders, C.; Gammerman, A.; and Vovk, V. 1999. Transduction with confidence and credibility. In *Proceedings of IJCAI'99*, volume 2, 722–726.

Schölkopf, B. 1997. *Support Vector Learning.* Ph.D. Dissertation, Max-Planck-Institut für biologische Kybernetik.

Totton, K. A. E., and Limb, P. 1991. Experience in neural networks for electronic diagnosis. In *2nd IEE Conference on Artificial Neural Networks.*

Vapnik, V. 1995. *The Nature of Statistical Learning Theory.* Springer.

Vapnik, V. N. 1998. *Statistical Learning Theory.* Wiley.