Pattern recognition based tools enabling autonomic computing.

Anton A. Bougaev
Purdue University
bougaev@purdue.edu

Abstract

Fault detection is one of the important constituents of fault tolerance, which in turn defines the dependability of autonomic computing. In presented work several pattern recognition tools were investigated in application to early fault detection. The optimal margin classifier technique was utilized to detect the abnormal behavior of software processes. The comparison with the performance of the quadratic classifiers is reported. The optimal margin classifiers were also implemented to the fault detection in hardware components. The impulse parameter probing technique was introduced to mitigate intermittent and transient fault problems. The pattern recognition framework of analysis of responses to a controlled component perturbation yielded promising results.

1. Introduction

In modern enterprise computer systems, pattern recognition finds a wide variety of applications. Various pattern recognition techniques are being applied to diagnostics of computer systems. However, each particular problem requires methodology refinement and accurate targeting. Unfortunately, there is no unifying framework for the solution of problems faced by system administrators in managing complex data centers. One of the examples of such problems is runaway process identification [1]. The overwhelming volume of information flows makes it virtually impossible for a system administrator to catch such system performance degradation precursors. A runaway process is a process that no longer provides its service and continues to use system resources. Runaway processes can cause dramatic performance degradation of multi-task, multi-user systems. This calls for the usage of a reliable autonomous detection method to assist the system administrator in identifying runaway processes and in taking appropriate actions.

2. Problem statement

The presented work has an objective to select and implement the most appropriate pattern recognition techniques both to software and hardware variable analysis. The motivation and experimental settings are described below.

2.1. Abnormal software process detection

To identify the state of a software process, snapshots of the state of variables associated with a given process are taken. The historic data is collected for a number of software processes of interest. At the next stage, the data points are being labeled by an expert, either manually or automatically, provided that a formal criterion of the process abnormality is given. The historic data is utilized to set up a classifier capable of discrimination of new unlabeled data points into two classes of normal and abnormal process states. Every data point is represented by as many as several hundred components thus having rather high dimensionality. To address this issue, an appropriate dimensionality reduction technique ought to be used. It is apparent, that not all the variables reported by the operating system's process statistics utility is relevant to determining if the process is a runaway. A feature extraction method needs deployed at this stage to make the volume of the data manageable in terms of computational resources needed to operate the classifier. The principal component analysis (PCA) can be utilized to significantly reduce the dimensionality of the input space. In this study, PCA is used as a basic reference technique for the dimensionality reduction procedure.

2.2. Hardware intermittent fault mitigation

Intermittent fault detection was another problem which was considered as a test bed in this study. The response of a set of hardware variables was used as a state representation data point for a hardware component. In this study the responses of a system board of an enterprise server was used in hope to discriminate at least some kinds of faults. The set of responses of known boards with intermittent faults was used along with a reference set of known fault free boards.

3. Pattern recognition tools

Once the training data was collected and labeled using expert judgment, the first approach implemented was a quadratic classifier. The quadratic classification has a solid theoretical foundation [2] and can be used as a reference technique during application of new methods under development.

The support vector method [3] was considered for the following reasons. In contrast to quadratic classification, where statistical properties of data distribution are subject to estimation, the support vector method broadly speaking is a non-parametric technique. Instead of parameter estimation, an optimal margin separating surface is to be built to separate two classes. The other beneficial feature is that support vector method is a kernel technique, which gives certain freedom in terms of incorporation of expert knowledge about invariance in the data. Potentially, this allows for the development a knowledge oriented classifier rather than one which is purely data-oriented.

4. Experimental results

The data from a process statistic utility was collected and its dimensionality was reduced using PCA. The number of the output principal components may vary. For illustration purposes, the reduction of dimensionality to two principal components was done and shown in Figure 1 schematically. The values on the axes do not have physical meaning since the information from many software variables was lumped. The typical behavior of a normal process is depicted by circles. An abnormal process (shown by triangles) initially was in the area where all the stable processes were. However, it has departed from normal behavior at a certain point in time. The difference of behavior with a normal process can be expressed in this case by principal components and the process can be recognized as a runaway process. The data represented in Figure 1 was used in a supervised design of a quadratic classifier and a support vector classifier. Both of the techniques have shown sufficient performance, which is acceptable for the automatic tracing of a runaway processes. In the case of hardware fault detection, the support vector machines have shown better performance. However, it was not possible to link features of physical responses of a system board to intermittent faults.

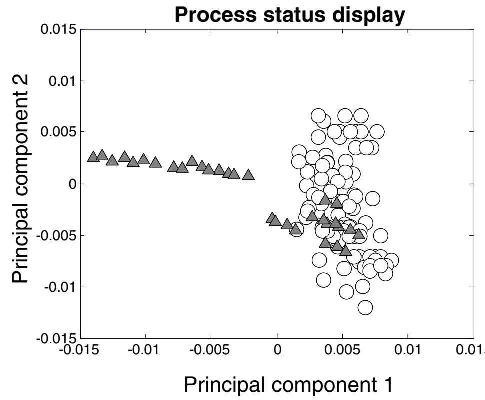

Figure 1. The process state representation after extraction of the principal components.

3. Conclusions

In the presented work, two feasible examples of the application of pattern recognition techniques were given. The prototyping carried out in the scope of this study shows that careful application of appropriate pattern recognition techniques is practically significant in the delivery of dependable computing. The incorporation of expert knowledge into the mechanism of classification allows for autonomous fault detection. The scoping experimentation resulted in practical guidelines for incorporation of pattern recognition engines into high-end enterprise computer systems.

4. Acknowledgements

The author wishes to thank Dr. Kenny Gross, Distinguished Engineer and Dr. Aleksey Urmanov, Researcher at Sun Microsystems, and Dr. Lefteri Tsoukalas, Head of the School of Nuclear Engineering at Purdue University for their helpful suggestions and feedback.

[1] K.S.Trivedi, K.Vaidyanathan, and K.Goseva-Postojanova, "Modeling and Analysis of Software Aging and Rejuvenation ", *Proc. 33rd Annual Simulation Symp.*, pp. 270-279, IEEE Computer Society Press, 2000.

[2] K. Fukunaga, *Introduction to statistical pattern recognition.* Academic press, Inc., Boston, USA, 2nd edition, 1990.

[3] V.Vapnik, *Statistical Learning Theory.* New York: A Wiley-Interscience Publication, 1998.