

# Fast Statistical Relationship Discovery in Massive Monitoring Data

Hui Zhang Haifeng Chen Guofei Jiang Xiaoqiao Meng Kenji Yoshihira  
NEC Laboratories America

Princeton, New Jersey 08540

Emails: {huizhang,haifeng,gfj,xqmeng,kenji}@nec-labs.com

**Abstract**—Today’s network systems are extensively instrumented for collecting a wealth of monitoring data. Statistical techniques like regression analysis can be applied to uncover rich relationships (e.g., correlation, causality, independence) between the measurement data which are further utilized for systems management tasks including fault diagnosis, configuration management, performance analysis, etc. However, one problem during this information mining process comes from the heavy computation overhead in statistical relationship discovery. In this paper, we propose a fast indexing technique to alleviate this problem by helping guide the discovery process in an optimal order. We model the optimal discovery process as the classic vertex cover problem in graph theory which is NP-complete. We use the heuristic of greedy vertex selection based on vertex degree and propose two simple algorithms for generating an estimated ranking (indexing) on the vertices (i.e., measurement points) based on the edges (the existing statistical relationships) incident to them. The two algorithms are based on random sampling and we analyze their output accuracy as a function of the sampling trials. On data traces from an operational 3G mobile network, our indexing technique performed close to the optimal solution (e.g., no more than 10% discovery time) and significantly better than random discovery (e.g., 70% less discovery time) on finding a specified percentage (e.g., 90%) of the existing relationships.

## I. INTRODUCTION

Large amounts of monitoring data can be collected from network system components for management issues. NetFlow, SNMP, “syslogs” are typical examples of such monitoring data. However, due to the dynamic nature of large-scale network systems and their complexity, it has been a great challenge to correlate the data effectively across different components and observation time. Statistical techniques like regression analysis have been applied to address this new problem as they can uncover rich pairwise relationships (e.g., linear or probabilistic correlation, causal relationships, data independence) among the measurement data which are further utilized for management tasks like infrastructure management and network traffic analysis. For example, Jiang et al [8] used AutoRegressive models with eXogenous inputs (ARX) to learn linear relationships between measurement data at different monitoring points, and then utilizes those relationships to build management tools including fault detection and capacity planning; Hua et al [6] built system performance analysis tool based on linear regression techniques; Qu et al [13] improved the prediction and accuracy of their anomaly detection algorithm for network security by computing the pairwise correlations between a large amount of features subtracted

from tremendous IP flow information; Huang et al [7] used PCA technique to diagnose network disruption with network-wide analysis.

However, one problem during this information mining process comes from the high computation overhead in statistical relationship discovery. For example, in [8], to decide if there is a simple linear relationship between the measured data from two monitoring points, a stream of data from each point has to be fitted into the regression model sequentially, then another stream of data has to be used for the model validation. Given the measurement data sets with size  $N$  (i.e., collected from  $N$  monitoring points), a full discovery requires  $N(N - 1)$  ( $\frac{N(N-1)}{2}$  for equivalent relationships) pairwise computations to discover all relationships that do exist. For a large system with hundreds or even thousands of monitoring points, the full-mesh computation may take hours to finish for one collection of monitoring data while a new collection of data is usually generated in the frequency of minutes continuously.

Two approaches can address this problem. One is to provide enough computing resources, and the other is to apply domain knowledge to cluster the measurement data sets and discover relationships only within individual clusters. The first approach may be the only solution if information loss is intolerable in the relationship discovery process. The second approach is efficient when such domain knowledge is available and accurate. However, our experiences with enterprise systems led to the observation that fine-grained data set clustering is hard to achieve due to the intrinsic dependency among system components, and information loss is usually tolerable within a certain degree due to the natural information redundancy contained in the relationships. Therefore, we propose a fast indexing technique to enable a third approach: guided relationship discovery which does not require manual pre-clustering and is computation cost and information loss controllable.

The guided relationship discovery picks the data sets in the decreasing order of the number of the relationships each has with the other sets, and therefore covers the complete relationship set in a fast way. Our indexing technique acts as an oracle by producing a rank of the data sets based on the relationship number. While the accurate ranking requires full-mesh computation, the fast indexing technique outputs an estimation whose accuracy is focused on the relative rank among data sets instead of individual sets’ absolute relationship number. Two algorithms for ranking estimation, both based on random

sampling, are proposed and we analyze their output accuracy as a function of the sampling trials. The experiment study with real-life data traces demonstrates both the effectiveness and efficiency of our indexing technique.

The rest of the paper is organized as following: Section II gives the problem definition and Section III presents the two randomized algorithms; the analysis results are shown in Section IV and the experiment results are presented in Section V; Section VI describes the related work, and Section VII concludes the paper with future work.

## II. PROBLEM DEFINITION

- **data set**
- **relationship**
- - - **relationship testing**

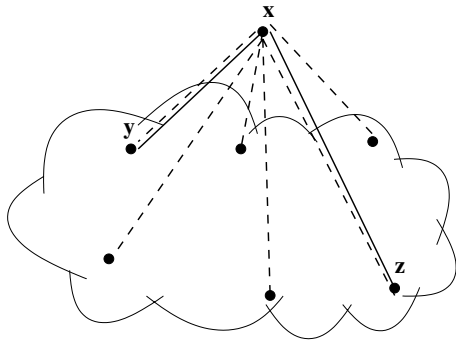


Fig. 1. A graph model of the relationship discovery process.

As shown in Figure 1, we model the pairwise relationship discovery process in massive data sets as a topology discovery problem in a graph  $G = (V, E)$ : a vertex represent one data set, and an edge exists between two vertices if the corresponding two data sets have the defined relationship. In the graph model, we consider a relationship is either 0 (non-existence) or 1 (existence). For statistical relationships with valued weight, we can map each value onto  $\{0, 1\}$  during the discovery process if the application is only interested in the relationships with a statistical significance (e.g., correlation coefficient larger than 0.5)<sup>1</sup>.

Initially, only the vertex set  $V$  is known and the edge set  $E$  is to be discovered. Without any information, a naive way to discover the topology is to pick the vertices with an arbitrary order, and at each time test (i.e., apply the relationship computation) the chosen vertex against the rest of the vertices to discover the edges. For example, in Figure 1  $x$  is chosen and two edges  $(x, y)$  and  $(x, z)$  are discovered after 6 relationship tests.

A better way is to pick the vertices in an optimal order to discover all edges within the minimal time. When the time to spend at each step is constant (e.g., testing the chosen vertex

<sup>1</sup>We consider symmetric relationships (undirected graphs) in this paper, but the extension of our technique to asymmetric relationships (directed graphs) is straightforward.

against all remaining vertices), the optimal solution is found if we have the minimal vertex set such that each edge is incident to a member of the set.

As the vertex cover problem is NP-complete [4], we apply a greedy heuristic which picks the vertex with highest degree at each step. While this heuristic can be off by a  $\ln(n)$  factor from optimum in the worst case and approximation algorithms with better worst-case performance are available in the literature, we choose it because it requires minimal topology information (vertex degree) and performs well in general even compared to those approximation algorithms [2].

There is one remaining problem before we apply the greedy algorithm: *how do we obtain the vertex degree information of an uncovered topology?* We solve this problem by sampling based estimation. As the discovery process is in the order of decreasing vertex degree, the estimation is focused on the relative ranking among the vertices instead of the absolute value of their degree. Next we present two rank estimation algorithms.

## III. RANK ESTIMATION ALGORITHM DESCRIPTION

### A. Algorithm 1: Uniform Sampling (US)

Given the computation budget of  $k$  relationship tests, the US algorithm runs as the following:

- 1) Keep one counter for each vertex. Initially all set to 0.
- 2) Randomly pick two different vertices  $x$  and  $y$  from the graph with uniform probability, apply a relationship testing. If the pair  $(x, y)$  has been picked before, the test is skipped and the cached result will be used for the next step.
- 3) If the test result is positive, increase the counters of  $x$  and  $y$  by 1.
- 4) Go back to 2 until  $k$  times.
- 5) Output a rank on all vertices based on the counter values; a tie is broken with a random choice.

In practice, the information of the test results and the sampled vertex pairs will be recorded for later usage in the guided relationship discovery to avoid repeated computations.

### B. Algorithm 2: Adaptive Sampling (AS)

Given the computation budget of  $k$  relationship tests, The AS algorithm runs as the following:

- 1) Keep one counter for each vertex. Initially all set to 1.
- 2) Randomly pick two different vertices  $x$  and  $y$  from the graph with probability proportional to their counter values, apply a relationship testing. If the pair  $(x, y)$  has been picked before, the test is skipped and the cached result will be used for the next step.
- 3) If the test result is positive, increase the counters of  $x$  and  $y$  by 1.
  - if the counter value of  $x$  ( $y$ ) is larger than a threshold (e.g., half of the vertex set size), we remove  $x$  ( $y$ ) from the vertex set in the following sampling process.
- 4) Go back to 2 until  $k$  times.

- 5) Output a rank on all vertices based on the counter values; a tie is broken with a random choice.

In practice, the information of the test results and the sampled vertex pairs will be recorded for later usage in the guided relationship discovery to avoid repeated computations.

### C. The Guided Relationship Discovery

The guided relationship discovery consists of three steps:

- 1) decide the computation budget for the rank estimation process which is either specified directly by the administrator or calculated based on the analysis results in Section IV.
- 2) apply one of the estimation algorithms to generate a vertex rank.
- 3) repeat picking one vertex in the order of the rank, testing its relationship with the remaining vertices, and removing it from the graph until either the overall computation budget is run out or the graph runs out of vertices.

## IV. ANALYSIS

### A. US Estimation Algorithm

For a vertex  $x$  with degree  $d_x$ , we define  $P_x$  as the probability that  $x$ 's counter is increased by 1 (i.e., one of  $x$ 's edges is discovered) at each sampling step.

*Property 1:* The US estimation is a Bernoulli process for  $x$  with

$$P_x = \frac{2d_x}{n(n-1)} \quad (1)$$

, where  $n$  is the vertex set size.

For the graph  $G$ , we define  $P_G$  as the probability that the testing result is positive (i.e., an edge is discovered) at one sampling step.

*Property 2:* The US estimation estimation is a Bernoulli process for  $G$  with

$$P_G = \frac{2m}{n(n-1)} \quad (2)$$

, where  $n$  is the vertex set size, and  $m$  is the edge set size.

For a vertex with degree  $d_x$ , let's denote the estimated degree output by the US algorithm as  $\hat{d}_x$ . Next we define a metric called Correct Ranking Probability (CRP) measuring the ranking accuracy between vertex  $x$  and vertex  $y$  whose degree is only a percentage  $\alpha_y$  of  $x$ 's ( $0 < \alpha_y < 1$ ):

$$CRP(d_x, \alpha_y, k) = Prob[\hat{d}_x > \hat{d}_y | d_x, d_y = \alpha_y d_x, k \text{ tests}] \quad (3)$$

For the US estimation algorithm, we can show

*Theorem 1:* For a vertex  $x$  with degree  $d_x$  and any vertex  $y$  whose degree is no more than  $\alpha d_x$ ,

$$CRP(d_x, \alpha_y, k) \geq \sum_{i=0}^n \sum_{j=i+1}^n \frac{e^{-\alpha\lambda} (\alpha\lambda)^i}{i!} \frac{e^{-\lambda} \lambda^j}{j!} \quad (4)$$

, where  $\lambda = \frac{2kd_x}{n(n-1)}$ ,  $k$  is the estimation trials, and  $n$  is the vertex set size.

*Proof:* Equation 4 comes from the Poisson approximation of the  $\hat{d}_x$  and  $\hat{d}_y$  distributions, and the decreasing property of CRP as a function of  $\alpha$ . We skip the details here. ■

Based on Theorem 1, an application administrator can calculate the required estimation budget with a specified estimation accuracy. For example, if the administrator wants to give discovery priority to any vertex having relationships with no less than 5% of the other vertices, Equation 4 can be used to compute the necessary  $k$  value to achieve a certain confidence (CRP).

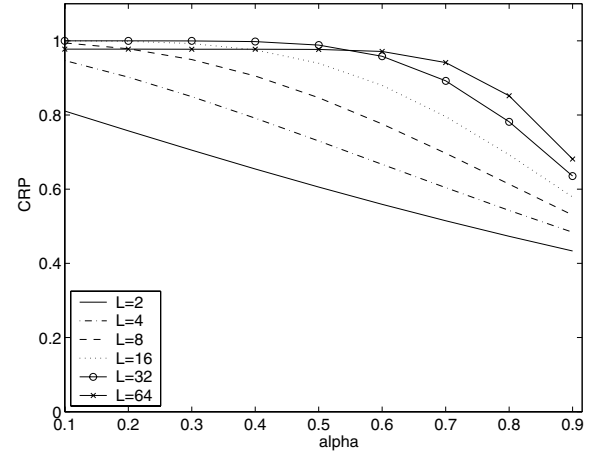


Fig. 2. The probability of correct ranking as a function of the degree difference  $\alpha$ .

Figure 2 shows the CRP as a function of  $\alpha$  given a fixed  $\lambda$  (2, 4, 8, 16, 32, 64) based on Theorem 1. For example, to make sure with probability 90% a vertex  $x$  is always ranked higher than any vertex whose degree is no larger than  $0.5d_x$ , we find the minimal  $\lambda$  is 8; therefore, the minimal estimation trials  $k \leq 80n$  if  $x$  refers to any vertex having relationships with no less than 5% of the other vertices.

Appendix contains a  $\langle \lambda, \alpha, CRP \rangle$  table which serves as a lookup table for application administrators to decide the estimation budget.

### B. AS Estimation Algorithm

The AS estimation is not a Bernoulli process for any vertex. The probability that a vertex has one edge discovered at a sampling trial is changing and dependent on the sampling history. We would like to point out that the derivative of a vertex's counter value over the estimation time is proportional to the summation of its neighboring vertices' counter values. This suggests that a vertex with a higher degree will have its probability of edge discovery keep increasing along with time compared to a vertex with a smaller degree. Therefore, the estimated rank would be more accurate compared to that by the US algorithm. Our experiment results validated this hypothesis.

## V. EXPERIMENT RESULTS

### A. Methodology

Our experiments are based on a collection of monitoring data from an operational UTRAN (UMTS Terrestrial Radio Access Network) system. A typical measurement report from the studied UTRAN system can be described as follows: every 15 minutes, 732 measurements are collected from over 500 cells. We further computed Key Performance Indicators (KPIs) from the raw measurement data using standard formulae. In the experiment, we had 129 KPI data sets for each monitoring period. The approach in [8] was used to test the correlation relationships between those KPI data sets; by choosing different correlation significance thresholds, different relationship graphs were generated on the same data set collection.

We ran the guided relationship discovery on the generated graphs: the computation budget for Step 1 was  $kn$  tests, where  $k = 4, 8$  and  $n = 129$ ; both US and AS algorithms were tested in Step 2; Step 3 was stopped when the graph ran out of vertices. The guided relationship discovery was compared with two other discovery processes; they were the same as Step 3 of the guided relationship discovery except that one used an arbitrary order of the vertices as the rank (called *random discovery*) and the other used the true rank of the vertices based on their degree (called *optimal discovery*). The metrics measured were the numbers of the vertices used in a discovery process to discover 80% and 90% of the edges.

### B. Results

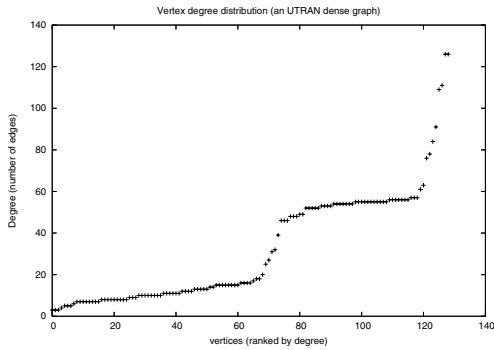


Fig. 3. The degree distribution of an UTRAN relationship graph (dense): avg degree = 32.6

1) *Dense Graphs*: The dense relationship graph in Figure 3 was generated with a low correlation significance threshold (0.5). It has an average degree of 32.6 for the 129-vertex graph. Figure 4 shows the edge discovery speed (in Step 3) of random discovery, optimal discovery, and our guided discovery with US algorithm and different estimation budget. On the dense graph, the US algorithm output a rank close to the true rank: the guided discovery with US algorithm and  $4n$  estimation tests covered 80% of the edges within 77 vertices, 90% of the edges within 96 vertices; with  $8n$  estimation tests, the guided discovery with US algorithm covered 80% of the edges within 64 vertices, 90% of the edges within 85 vertices; the optimal

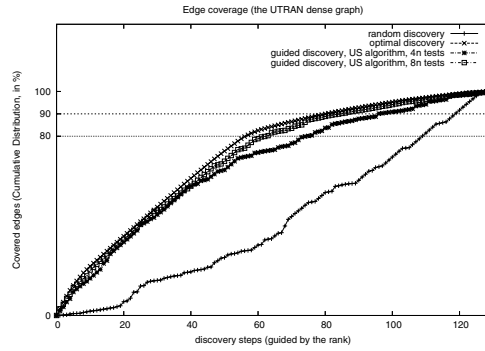


Fig. 4. US estimation algorithm, UTRAN dense graph.

discovery required 57 (81) vertices to cover 80% (90%) edges; the random discovery required 110 (119) vertices to cover 80% (90%) edges.

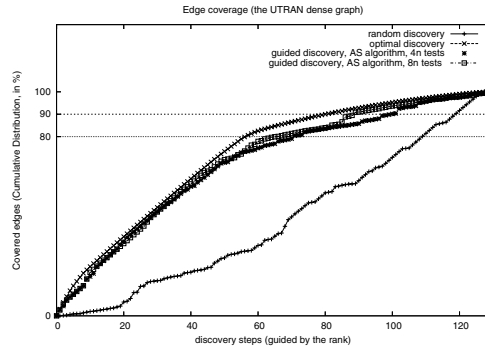


Fig. 5. AS estimation algorithm, UTRAN dense graph.

We also tested the AS algorithm on the dense graph. Similar to the US algorithm, the AS algorithm helped the guided discovery with a near-optimal rank: with  $4n$  ( $8n$ ) estimation tests, it covered 80% of the edges within 72 (66) vertices, 90% of the edges within 100 (89) vertices.

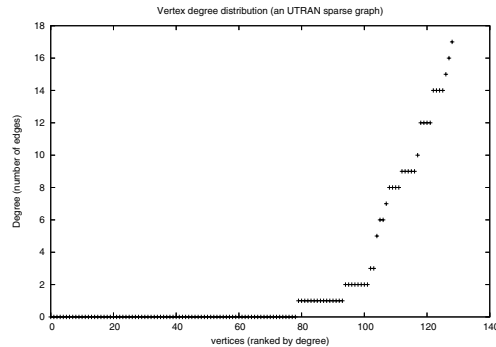


Fig. 6. The degree distribution of a UTRAN relationship graph (sparse): avg degree = 2.3

2) *Sparse Relationship Graph*: The sparse relationship graph in Figure 6 was generated with a high correlation significance threshold (0.9). It has an average degree of 2.3. Figure 7 shows the edge discovery speed (in Step 3) of

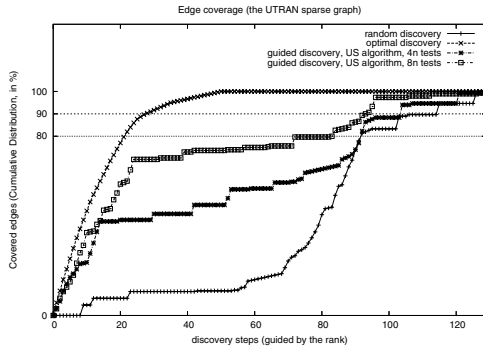


Fig. 7. US estimation algorithm, UTRAN sparse graph.

random discovery, optimal discovery, and our guided discovery with US algorithm and different estimation budget. On the sparse graph, the US algorithm output a rank away from the true rank with a limited estimation budget: the guided discovery with US algorithm and  $4n$  estimation tests covered 80% of the edges within 92 vertices, 90% of the edges within 104 vertices; with  $8n$  estimation tests, the guided discovery with US algorithm covered 80% of the edges within 72 vertices, 90% of the edges within 93 vertices; the optimal discovery required 22 (28) vertices to cover 80% (90%) edges; the random discovery required 92 (106) vertices to cover 80% (90%) edges.

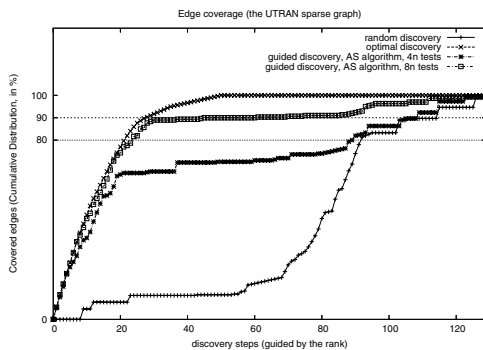


Fig. 8. AS estimation algorithm, UTRAN sparse graph.

However, the AS algorithm still performed well on the sparse graph: with  $8n$  estimation tests, it covered 80% of the edges within 24 vertices, and covered 89% of the edges within the 30 vertices whose estimation counters had non-zero values.

## VI. RELATED WORK

New system management techniques based on statistical analysis were developed recently [8], [6], [5], [13]. They effectively utilized the massive monitoring data collection available in the studied systems by correlating the data across different components and observation time, and applying appropriate statistical models to catch the fundamental features of the complex systems.

Sampling technique has been applied for efficient data mining. For example, Mamitsuka et al [10] and Kollios et al [9] proposed different biased sampling methods to speed up the operation of general data mining tasks such as clustering and outlier detection in large multi-dimensional data sets. Wu and Jermaine [12] proposed a random sampling algorithm for distance-based outlier detection in high-dimensional data sets.

In graph theory, topology discovery on vertices is a well-studied problem where the vertex set is to be discovered and the edges incident to a vertex are *automatically* known once it is visited. Cooper and Frieze [3] studied the discovery time of random walking on sparse random graphs while Adamic et al [1] shows the optimality of biased walking based on vertex degree in power-law random graphs. Thompson [11] gave a survey on link-tracing based adaptive sampling in graphs.

## VII. CONCLUSIONS & FUTURE WORK

In this paper, we studied the problem of pairwise relationship discovery in massive monitoring data for network management. Two sampling based rank estimation algorithms along with a guided discovery scheme were proposed to enable fast and approximate relationship discovery.

Preliminary results were presented on the analysis and evaluation of the two algorithms. We are investigating the property of the adaptive sampling algorithm in terms of the correct ranking probability. Experiments on more real-world data and synthetic topologies (e.g., power-law random graphs) will be done. Two other issues are also interesting to be studied: the savings on the computation overhead with the fast indexing technique on arbitrary graphs and graphs with special properties (e.g., edges representing equivalent relationships); the optimal allocation between estimation time and discovery time in the guided discovery process.

## APPENDIX

TABLE I  
THE CORRECT-RANKING PROBABILITY TABLE

$\lambda$	$\alpha$	CRP	$\lambda$	$\alpha$	CRP
1.0	0.1	0.596242	1.0	0.3	0.529754
1.0	0.5	0.469870	1.0	0.7	0.416082
1.0	0.9	0.367893	5.0	0.1	0.970049
5.0	0.3	0.887477	5.0	0.5	0.768692
5.0	0.7	0.633181	5.0	0.9	0.498783
10.0	0.1	0.997916	10.0	0.3	0.969373
10.0	0.5	0.880206	10.0	0.7	0.728135
10.0	0.9	0.545258	15.0	0.1	0.999834
15.0	0.3	0.990727	15.0	0.5	0.932594
15.0	0.7	0.786495	15.0	0.9	0.574093
20.0	0.1	0.999985	20.0	0.3	0.997060
20.0	0.5	0.960654	20.0	0.7	0.828004
20.0	0.9	0.596011	25.0	0.1	0.999995
25.0	0.3	0.999040	25.0	0.5	0.976536
25.0	0.7	0.859364	25.0	0.9	0.614095
30.0	0.1	0.999973	30.0	0.3	0.999656
30.0	0.5	0.985786	30.0	0.7	0.883835

## REFERENCES

- [1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Phys. Rev. E*, 64(4):046135, Sep 2001.
- [2] C. Baird, R. Rockstroh, and A. Parrish. *Vertex Covers - Trials and Tribulations*. U. of Maryland CMSC451 class project report, <http://www.cs.umd.edu/class/summer2006/cmssc451/>, 2006.
- [3] C. Cooper and A. Frieze. The cover time of sparse random graphs, 2003.
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, Second Edition, 2001.
- [5] Z. Guo, G. Jiang, H. Chen, and K. Yoshihira. Tracking probabilistic correlation of monitoring data for fault detection in complex systems. In *DSN '06: Proceedings of the International Conference on Dependable Systems and Networks (DSN'06)*, pages 259–268, Washington, DC, USA, 2006. IEEE Computer Society.
- [6] K. A. Hua, N. Jiang, R. Villafane, and D. Tran. Admire: an algebraic approach to system performance analysis using data mining techniques. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 490–496, New York, NY, USA, 2003. ACM Press.
- [7] Y. Huang, N. Feamster, A. Lakhina, and J. Xu. Detecting network disruptions with network-wide analysis. In *Proceedings of the 2007 ACM SIGMETRICS*, June 2007.
- [8] G. Jiang, H. Chen, and K. Yoshihira. Discovering likely invariants of distributed transaction systems for autonomic system management. *Cluster Computing*, 9(4):385–399, 2006.
- [9] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold. Efficient biased sampling for approximate clustering and outlier detection in large data sets. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1170–1187, 2003.
- [10] H. Mamitsuka and N. Abe. Efficient mining from large databases by query learning. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 575–582, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [11] S. K. Thompson. Adaptive sampling in graphs. In *Proceedings of the Survey Research Methods Section, American Statistical Association*, 1998.
- [12] M. Wu and C. Jermaine. Outlier detection by sampling with accuracy guarantees. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 767–772, New York, NY, USA, 2006. ACM Press.
- [13] T. Xia, G. Qu, S. Hariri, and M. Yousif. Genetic algorithm and information theory - a hybrid approach on intrusion detection. in *Proceedings of 24th IEEE International Performance Computing and Communications Conference*, 2005.