Active Sampling Approaches in Systems Management Applications

Irina Rish
IBM T.J. Watson Research
Hawthorne, NY 10532, USA
Email: rish@us.ibm.com

***Abstract*— Systems management tasks such as problem diagnosis and resource allocation often require making quick, real-time inferences using available systems measurements, and smart choice of such measurements can greatly improve the quality and speed of decision-making. On one hand, systems management domain has certain advantage over some other applications of machine-learning since it often allows active approaches to data collection (which might be very expensive or even impossible in other, "natural" applications such as, for example biology and medicine). On the other hand, given an enormous number of possible measurements, selecting the minimal number of most-relevant ones, either ahead of time or adaptively, is recognized as a difficult systems management problem often addressed in an ad-hoc way ("we are drowning in data but starving for knowledge"). In this paper, we provide a summary of our approaches on active sampling in distributed computer systems, that include our "active probing" approach for problem diagnosis and active learning for end-to-end performance prediction and resource allocation in distributed systems and networks. We present successful empirical results demonstrating advantages of our approaches in several practical applications.**

I. Introduction

As distributed computer systems continue to grow in their size and complexity, systems management becomes more and more challenging and requires scalable and cost-efficient approaches capable of making real-time inferences from huge volumes of various system events and measurements. Particularly, selecting a small subset of most-relevant measurements is often necessary, especially when the amount of all possible measurements is extremely large. Systems management domain is naturally suited for using active, real-time data collection approaches that we refer to as *active sampling*, as opposed to standard "passive" data analysis.

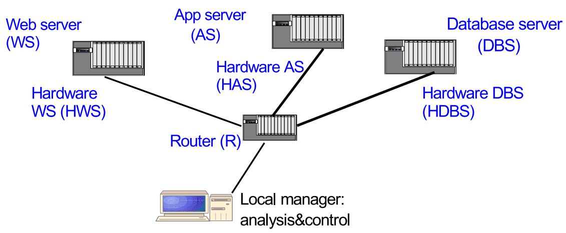
In this paper, we demonstrate advantages of active sampling approaches in two systems management tasks: problem diagnosis and resource allocation. First, we summarize our recent work on *active probing* for problem diagnosis presented in [8], [18], [13], and then present our current work on active learning in resource allocation domain using collaborative filtering framework. Note that our methods are quite general and can be applied in other domains besides systems management.

Our first application is problem diagnosis in distributed systems and network. For example, in IP network management, we would like to quickly identify which router or link has a problem when a failure or performance degradation occurs in the network. In the e-Commerce context, our objective could be to trace the root-cause of unsuccessful or slow user transactions (e.g. purchase requests sent through a web server) in order to identify whether it is a network problem, a web or back-end database server problem, etc. Another example is real-time monitoring and performance diagnosis in a large cluster or Grid system containing hundreds or thousands of computers.

We propose an adaptive diagnostic approach called *active probing*, that allows a fast online inference about the current system state via active selection of only a small number of most-informative measurements called *probes*. A probe is a test transaction (such as ping, traceroute, webpage access, database query, an e-commerce transaction, etc.) whose outcome depends on some of the system's components. A probe can be viewed as a disjunctive test over the components involved in the probe: indeed, a probe is OK if and only if all the involved components are OK, otherwise the probe fails. [1] In case of noisy probe outcomes, we formulate diagnosis as an inference problem in a *noisy-OR* Bayesian network that generalizes disjunctive testing problem. Active probing is an information-theoretic approach that dynamically selects probes maximizing the *information gain* about the current system state. Empirical results on both simulated and real applications show that active probing requires on average much less probes than previously used pre-planned probing (when the set of measurements is fixed in advance), sometimes reducing the probe set size by $60-70\%$.

Next, we consider a resource allocation application - selecting a service provider in a distributed environment - where active sampling promises to be quite effective. For example, given a large number of available servers that can provide some service to a client (e.g., file download in a data Grid, or computational resources in a computational Grid, etc.), the problem is to select "the best" server(s) for a given request. We focus on the problem of predicting the

[1] The problem of selecting minimal number of probes for diagnosis is closely related to the *group testing* problem [3] but is more complex due to constraints on probe construction, such as the network topology, and available application-level transactions.



Dependency matrix

Probe types \ Nodes	WS	AS	DBS	R	HWS	HAS	HDBS
pWS	1	1	1	1	1	1	1
pAS	0	1	1	1	0	1	1
pDBS	0	0	1	1	0	0	1
pingR	0	0	0	1	0	0	0
pingWS	0	0	0	1	1	0	0
pingAS	0	0	0	1	0	1	0
pingDBS	0	0	0	1	0	0	1

Probes:
pWS - Web Page access
pAS- Application Server access
pDBS - Database query
pingR – ping router
pingWS – ping Web Server
pingAS - ping Application server
pingDBS - ping Database Server

Fig. 1. (a) A simple benchmark distributed system with one probe station and 7 probes; (b) dependency matrix for the system in (a).

quality of service that the given server could provide to a client based on the whole history of pairwise interactions between all clients and all servers in the past. Typically, the interaction matrix is quite sparse, so the problem is closely related to collaborative filtering. We apply the state-of-art matrix factorization technique for collaborative filtering and augment it with active sampling. We demonstrate empirical results on the bandwidth data for IBM's download Grid (a Grid-like file distribution system) and for PlanetLab latency data. Active sampling shows promising improvement in prediction accuracy over the random sampling approach.

II. Background and Definitions

Suppose we wish to monitor a system of networked computers. Let $\mathbf{X}$ represent the binary state of N network elements. $X_i = 0$ indicates that the element is in normal operation mode, and $X_i = 1$ indicates that the element is faulty. We can take X_i to be any system component whose state can be measured using a suite of tests. If the system is large, it is often impossible to test each individual component directly. A common solution is to test a subset of components with a single *test probe*. If all the test components are okay, the test would return a 0. Otherwise the test would return 1, but it does not reveal which components are faulty.

We assume there are machines designated as *probe stations*, which are instrumented to send out *probes* to test the response of the network elements represented by $\mathbf{X}$. Let $\mathbf{T}$ denote the available set of probes. A probe can be as simple as a *ping* request, which detects network availability. A more sophisticated probe might be an e-mail message or a webpage-access request.

Example 1: Figure 1 shows an example of a simple benchmark system that contains a probing station, which sends various probes through a common router to a Web Server (WS), Application Server (AS), and Database Server (DBS). Note that WS, AS and DBS represent the OK/not OK state of the corresponding applications running on these machines, while HWS, HAS and $HDBS$ denote "hardware" problems with WS, AS, and DBS, respectively (in our case, HWS is OK if WS can be reached by $ping$ command; however, the web server application may not be running, and thus WS is not OK). R will denote the state of the router. Also, Figure 1 shows the "dependency matrix" between system's states and the following set of probes, where the matrix entry (ij) equals 1 if and only if the outcome of probe i depends on component j:

- "main" probe called pWS, attempts to open a web page on WS, which also runs an application on AS, which in its turn sends a query to a database on DBS. The outcome of this probe depends on the state (i.e., OK/not OK) of all components, i.e. WS, HWS, AS, HAS, DBS, and $HDBS$, as well as on the state of the router R. Thus, the row of the probe pWS contains ones in all columns (i.e., fails if any of these components fail).
- probe pAS calls an application on AS which sends a query to the database on DBS; thus the probe depends on the states of AS, HAS, DBS, $HDBS$, R.
- probe $pDBS$ sends a query to the database on DBS, and thus depends on DBS, $HDBS$ and R.
- probes $pingR$, $pingWS$, $pingAS$ and $pingDBS$ are simply $ping$ commands to the router and the corresponding servers.

In the absence of noise a probe is a disjunctive test: it fails if an only if there is at least one failed node on its path. More generally, it is a noisy-OR test [12]. The joint PDF of all tests and network nodes forms the well-known QMR-DT model [9], a bipartite noisy-OR *Bayesian network* model where the top-level nodes (X_i's) are hidden and the bottom-level nodes (T_j's) are observed test outcomes:

$$P(x_j) = (\alpha_j)^{x_j}(1-\alpha_j)^{(1-x_j)}, \tag{1}$$

$$P(t_i = 0 \mid \mathbf{s}_{\mathbf{pa}_i}) = \rho_{i0} \prod_{j \in \mathbf{pa}_i} \rho_{ij}^{x_j}, \tag{2}$$

$$P(\mathbf{x}, \mathbf{t}) = \prod_i P(t_i \mid \mathbf{s}_{\mathbf{pa}_i}) \prod_j P(x_j). \tag{3}$$

Here, $\alpha_j := P(x_j = 1)$ is the prior fault probability, ρ_{ij} is the so-called inhibition probability, and $(1-\rho_{i0})$ is the leak probability of an omitted faulty element. The inhibition probability is a measurement of the amount of noise in the network.

III. Diagnosis with Test Selection

In many diagnosis problems, the user has an opportunity to select tests in order to improve the accuracy of diagnosis. For example, in medical diagnosis, doctors face the *experiment design* problem of choosing which medical tests to perform next.

Our objective is to maximize diagnostic quality while minimizing the cost of testing. The diagnostic quality of a subset of tests $\mathbf{T}^*$ can be measured by the amount of uncertainty about $\mathbf{X}$ that remains after observing $\mathbf{T}^*$. From the information-theoretic

Active Probing

Input: A set of available probes $\mathbf{T}$ and a prior distribution over system states $P(\mathbf{X})$.
Output: A set $\mathbf{T}_a$ of probes and their outcomes, posterior distribution $Belief(\mathbf{X})$ and its support $\mathbf{S}$.
Initialize: $Belief(\mathbf{X}) = P(\mathbf{X})$, $\mathbf{T_a} = \emptyset$,
$\mathbf{S}$ = support of $P(\mathbf{X})$.
do
1. select current most-informative probe: $Y^* = \arg\max_{Y \in \mathbf{T} \setminus \mathbf{T_a}} I(\mathbf{X}; Y|\mathbf{T_a})$
2. execute Y^*; it returns $Y^* = y^*$ (0 or 1)
3. update $\mathbf{T}_a = \mathbf{T}_a \cup \{Y^* = y^*\}$
4. update $Belief(\mathbf{X}) = P(\mathbf{X}|\mathbf{T_a})$

while $\exists Y \in T$ such that $I(\mathbf{X}; Y|\mathbf{T}_a) > 0$
Return $\mathbf{T_a}$, $Belief(\mathbf{X})$, S =support of $Belief(\mathbf{X})$.

Fig. 2. Active Probing algorithm for probabilistic diagnosis with most-informative probe selection.

	# of nodes	# of probes	Pre-planned probes (exact)	Pre-planned probes (greedy)	Pre-planned probes (detect)	Active Probing: min	Active Probing: max	Active Probing: average	Savings % active vs exact
O1	24/19	22/22	15	15	8	3	8	4.9	67%
O2	43/38	44/32	27	27	17	3	17	7.7	72%
O3	40/34	29/29	24	24	16	3	16	7.5	69%
G1	10/8	14/8	5	5	3	3	4	3.2	36%
C1	149/50	27/27	27	27	27	4	27	10.8	60%
C2	122/46	27/27	24	24	24	4	24	9.8	61%
C3	56/32	27/23	17	17	14	4	13	6.7	71%
C4	34/23	27/22	12	13	10	3	10	5.5	75%

Fig. 3. Active probing results on several practical problems.

perspective, a natural measurement of uncertainty is the conditional entropy $H(\mathbf{X} \mid \mathbf{T}^*)$. Clearly, $H(\mathbf{X} \mid \mathbf{T}) \leq H(\mathbf{X} \mid \mathbf{T}^*)$ for all $\mathbf{T}^* \subseteq \mathbf{T}$. Thus the problem is to find $\mathbf{T}^* \subseteq \mathbf{T}$ which minimizes both $H(\mathbf{X} \mid \mathbf{T}^*)$ and the cost of testing. When all tests have equal cost, this is equivalent to minimizing the number of tests.

This problem is known to be NP-hard [8]. A simple greedy approximation is to choose the next test to be $T^* = \arg\min_T H(\mathbf{X} \mid T, \mathbf{T}')$, where $\mathbf{T}'$ is the currently selected test set. The expected number of tests produced by the greedy strategy is known to be within a $O(\log N)$ factor from optimal (see [18]). The same result holds for approximations (within a constant multiplicative factor) to the greedy approach. Furthermore, our empirical results show that the approach works well in practice [8].

We make a distinction between nonadaptive (off-line) and adaptive (online) test selection. In online selection, previous test outcomes are available when selecting the next test. Off-line test selection attempts to plan a suite of tests before any observations have been made. We will focus on the adaptive approach which is typically much more efficient in practice than its off-line counterpart [8].

Adaptive Test Selection Problem: Given the observed outcome $\mathbf{t}'$ of previously selected sequence of tests $\mathbf{T}'$, select the next test to be $\arg\min_T H(\mathbf{X} \mid T, \mathbf{t}')$.

In the context of our probing scenario, we propose the *active probing* algorithm, outlined in Figure 2, that iterates between requesting next most-informative probe and updating the posterior distribution ("belief") over the states of system components.

A. Empirical Results

In Table 3, we compare active probing versus off-line probe selection on several real probing applications and demonstrate significant savings in the number of probes. The problem $G1$ is a relatively small testbed for probe analysis, while the sets of problems $O1-O3$ and $C1-C4$ relate to networks supporting several different e-business applications, which include many servers and routers, and its performance and availability depends on a large number of software components. An initial set of probes was manually selected by an expert for the case of single fault localization. We ran a simple initial preprocessing on the dependency matrix in order to eliminate repeating probes and merge indistinguishable nodes, i.e. the nodes whose faults are indistinguishable given the original probe set. The first two column of Table 3 show the number of nodes and probes before and after the initial preprocessing. The next two columns show the minimum number of probes found in a pre-planning phase by exhaustive and by greedy search, respectively. Then the next column shows the minimum number of probes (found in greedy way) necessary for fault detection only (i.e., simply the probes 'covering' all nodes). Finally, we show the minimum and the maximum number of probes required by active probing to diagnose a single fault, and the average such number over all possible faults. In most of the cases (except for a small testbed problem $G1$), active probing was saving from 60% to 75% probes if compared to pre-planned probing (and even more if compared to the initial probe set size).

B. Computational Complexity Issues

Online greedy approach for maximizing information gain described above is quite efficient when the number of faults is small and thus the state space of unobserved variables can be easily enumerated. However, in a general multi-fault case the state space is exponential in the number of variables, and a compact representation such as Bayesian network must be used. Unfortunately, exact computation of conditional entropies in a general Bayesian network is also intractable [18]. Most of the existing literature on value of information and most-informative test selection [7],

[2], [6], [13] does not seem to focus on the computational complexity of most-informative test selection in a general Bayesian network setting, except for the most recent work by [10]. However, [10] focus on the nonadaptive ("non-myopic") problem and provide algorithms for efficient selection of a most-informative test subset given a bound on its size. Our problem is different as we consider adaptive ("myopic") test selection without a particular bound on the number of tests. In [18], we proposed an approximation algorithm for computing marginal conditional entropy that is based on modification of loopy belief propagation, a popular approximate inference method in Bayesian networks, and illustrate promising empirical results for problem diagnosis in large-scale, realistic Internet-like computer networks using INET network generator [17].

IV. Active Collaborative Prediction for End-to-End Performance Estimation and Resource Allocation (Work in Progress)

There are various other systems management applications where it is too costly or impossible to obtain all desirable measurements and active sampling may prove to be useful.

For example, it is often important to estimate latency, bandwidth, or other end-to-end performance metrics in large-scale networks where we cannot simply make direct measurements between all pairs of nodes, and would like to make predictions about unobserved metrics from the observed ones. There were multiple previous attempts to solve this problem, each making various assumptions about the performance data. For example, Global Network Positioning (GNP) system [4] assumes that the network nodes can be embedded into a low-dimensional Euclidean space, and the latency between the nodes is viewed as a distance function. This is not a very natural assumption (e.g., triangle inequality does not necessarily holds for latencies), and the embedding algorithm is known to converge slowly and depend on the initial assignment. Another approach – the algorithm called Vivaldi [1] – makes an analogy with a physical network connected by springs, and tries to place the nodes so that the potential energy of the system is minimized (which is again a questionable assumption for modeling network latencies, bandwidth etc). Yet another type of approaches uses so-called Lipschitz embeddings into an N-dimensional space where a node coordinates are given by its distances to a set of N *landmark* nodes, and hosts with similar distances to landmark nodes are close to each other in the space – again, a questionable assumption, since having similar latency from nodes A and B to node C does not necessarily imply low latency between A and B in an arbitrary network. Lipshitz embeddings are further combined with PCA to reduce the dimensionality of the space [5], [15].

Recently, a matrix factorization approach was proposed for latency prediction, particularly based on singular value decomposition (SVD) and non-negative matrix factorization (NMF) techniques [11]. We improve upon this approach in several ways. First, we use a more advanced max-margin matrix factorization (MMMF) method recently proposed by [14] which tends to give a better prediction accuracy. Second, and more important, contribution is using an active sampling on top of MMMF approach that leads to a noticeable further improvement in prediction accuracy. Finally, we go beyond just matrix factorization approaches and propose to use a more general *collaborative prediction* formulation as a rich source of prediction techniques, that include, besides matrix factorizations, such approaches as clustering and probabilistic graphical models.

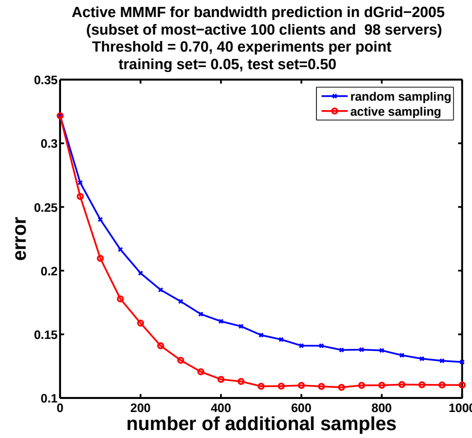
Accurate prediction of end-to-end performance can be used further used for various management tasks, such as network monitoring, diagnosis, and resource allocation, just to name a few. For example, such knowledge can be used for better service provider selection in Grid computing, optimal selection of content provider in a content-distribution system (e.g., fast real-time file, movie or music download), and so on. A particular application we encountered in our practice was an IBM-internal peer-to-peer content distribution systems for file download, called *downloadGrid*. DownloadGrid architecture has some similarities with the Internet-based Gnutella, Napster and BitTorrent file distribution systems as it allows peer-to-peer file downloading; however, it combines the peer-to-peer approach with centralized decision-making architecture for matching "clients" and "servers". Centralized architecture is mainly motivated by security issues, but can also provide opportunities for optimization of the overall system's performance: for example, it allows to collect system-wide historic data about the previous file downloads which can be used later for predicting the end-to-end performance for previously unobserved client-server pairs, and for (nearly) optimal selection of a server(s) for a particular client file request.

Formally, the problem can be stated as follows: given a set of clients (peers that might request a service, such as file download), a set of servers (peers that can provide a service, e.g. have files of interest), and historic data measuring some performance metric for each client-server interaction (e.g., bandwidth), we wish to predict the performance of (ideally, all) servers with respect to the given client and choose the best server. If we simplify this problem by using some aggregate metric for all interactions between a particular client-server pair, the data can be represented by a matrix where rows correspond to clients, columns correspond to servers, and the matrix entries represent a metric (e.g., average bandwidth) characterizing the (average) quality of a particular client-server interaction. (Note that the matrix can be extremely sparse: e.g., in some of our datasets, less than 1% of the matrices were filled). The problem becomes practically equivalent to collaborative filtering problem extensively studied in the literature.

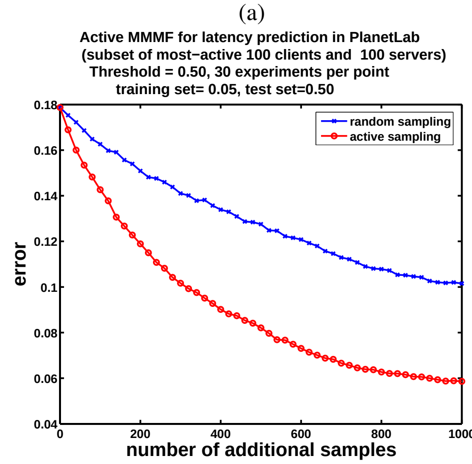
As mentioned above, our approach is based on collaborative prediction using max-margin matrix factorization (MMMF) method [14], which we do not describe here in detail due to lack of space. Briefly, given an incomplete matrix Y, the matrix factorization problem is to find an approximation $X = UV$ to Y. MMMF views U as a set of feature vectors, and V as a set of linear classifiers, so that search for "best" X can be viewed as a simultaneous search for a set of bounded-norm feature vectors and a set of max-margin linear classifiers, which turns out to be a convex optimization problem, unlike the previous fixed-rank (rather than fixed-norm) SVD-like approaches. This interpretation of MMMF makes it easy to introduce active sampling, i.e. any heuristic for active learning of SVMs can be potentially added on top of MMMF. We use the active learning heuristic proposed by Tong and Koller [16] that prefers the *minimum-margin* sample (i.e. the one we are least confident about).

We present promising empirical results in FigureBN, for downloadGrid (a) and for PlanetLab latency data (b). In both cases, we use a threshold (e.g., median) to convert real-valued bandwidth data to binary ones, where -1 and 1 labels mean below and above the threshold ("good" and "bad"), respectively. We selected a subsets of about 100 clients and 100 servers that had maximum number of interactions. We started with initial training set which contained only 5% of non-zero entries in each of the matrices, and set aside a test set containing 50% of the non-zero entries. The rest was used as a pool for sampling. The Y axis shows the prediction accuracy, while the X-axis shows the number of additional samples selected either by active or random sampling. We observe that active sampling results into consistently more accurate predictions than the random sampling using same number of samples.

Using more advanced active learning heuristics, as well as extending our approach to probabilistic collaborative prediction methods (e.g., aspect model, MCVQ, etc) remain the directions for future work.



(a)



(b)

Fig. 4. Active vs. random sampling for predicting (a)download Grid bandwidth and (b) PlanetLab latency.

References

[1] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *Proc. of ACM SIGCOMM 2004, Aug 2004*.

[2] J. de Kleer and B.C. Williams. Diagnosing Multiple Faults. *Artificial Intelligence*, 32(1), 1987.

[3] D-Z. Du and F.K. Hwang. *Combinatorial Group Testing and Its Applications (2nd edition)*. World Scientific, 2000.

[4] T.S. Eugene and Hui Zhang. Predicting Internet Distance with Coordinate-Based Approaches. In *Proc. of INFOCOM 2002, New York City, NY, June 2002*.

[5] C. Choi H. Lim, J. Hou. Consructing Internet Coordiante System Based on Delay Measurement. In *Proc. of IMC 2003, Oct 2003*.

[6] D. E. Heckerman, E. J. Horvitz, and B. Middleton. An approximate nonmyopic computation for value of information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:292–298, 1993.

[7] R. Howard. Information value theory. *IEEE Trans Syst Sci Cybern*, 2(1):22–26, 1966.

[8] S. Ma N. Odintsova A. Beygelzimer G. Grabarnik K. Hernandez I. Rish, M. Brodie. Adaptive diagnosis in distributed systems. *IEEE Transactions on Neural Networks (special issue on Adaptive Learning Systems in Communication Networks)*, 16(5):1088–1109, 2005.

[9] T. Jaakkola and M. Jordan. Variational probabilistic inference and the qmr-dt database. *Journal of Artificial Intelligence Research*, pages 291–322, 1999.

[10] A. Krause and C. Guestrin. Near-optimal Nonmyopic Value of Information in Graphical Models. In *Proc. of Uncertainty in Artificial Intelligence (UAI)*, 2005.

[11] Y. Mao and L. K. Saul. Modeling Distances in Large-Scale Networks by Matrix Factorization. In *Proc. of IMC 2004, Oct 2004*.

[12] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Mateo, California, 1988.

[13] I. Rish, M. Brodie, N. Odintsova, S. Ma, and G. Grabarnik. Real-time Problem Determination in Distributed Systems using Active Probing. In *Proceedings of 2004 IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), Seoul, Korea*, pages 133–146, 2004.

[14] Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum Margin Matrix Factorizations. In *Advances in Neural Information Processing Systems (NIPS-04)*, 2004.

[15] L. Tang and M. Crovella. Virtual Landmarks for the Internet. In *Proc. of IMC 2003, Oct 2003*.

[16] S. Tong and D. Koller. Support Vector Machine Active Learning with Applications to Text Classification, booktitle=.

[17] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report CSE-TR-456-02, University of Michigan, 2002.

[18] A. Zheng, I. Rish, and A. Beygelzimer. Efficient Test Selection in Active Diagnosis via Entropy Approximation. In *Proceedings of UAI-05*, 2005.