The Price of Incorrectly Aggregating Coverage Values in Sensor Selection

Amotz Bar-Noy*†, Matthew P. Johnson^{‡†}, Nooreddin Naghibolhosseini[†], Dror Rawitz[§] and Simon Shamoun[§]

*Brooklyn College

†The CUNY Graduate Center

[‡]Lehman College

§Bar Ilan University

Abstract—An important problem in the study of sensor networks is how to select a set of sensors that maximizes coverage of other sensors. Given pairwise coverage values, three commonly found functions give some estimate of the aggregate coverage possible by a set of sensors: maximum coverage by any selected sensor (MAX), total coverage by all selected sensors (SUM), and the probability of correct prediction by at least one sensor (PROB). MAX and SUM are two extremes of possible coverage, while PROB, based on an independence assumption, is in the middle. This paper addresses the following question: what guarantees can be made of coverage that is evaluated by an unknown sub-modular function of coverage when sensors are selected according to MAX, SUM, or PROB? We prove that the guarantees are very bad: In the worst case, coverage differs by a factor of sqrt(n), where n is the number of sensors. We show in simulations on synthetic and real data that the differences can be quite high as well. We show how to potentially address this problem using a hybrid of the coverage functions.

I. INTRODUCTION

An ongoing research topic in sensor networks is how to use the data from a limited number of sensors to predict the data of all remaining sensors [1]–[3]. Given a function of the prediction quality by a set of sensors, the problem is to select the set of sensors under some budget constraints that maximizes the total prediction quality. This problem is closely related to the sensor placement problem [4]–[8], which is to select a subset of locations that best cover all locations given a function of sensor coverage. Coverage in these cases is typically defined as the probability of detecting events at these locations rather than estimating sensor values.

Such problems are typically NP-hard, and approximation guarantees are not always possible. According to most formulations, the coverage function is sub-modular [1], [3], [9], which means the sum of the coverage by two different sets is no less than the prediction quality by their union. An example

This work is supported by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053, The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. Matthew P. Johnson was supported in part by a PSC-CUNY Grant (no. 67665-00) and a CUNY Collaborative Incentive Research Grant (no. 2153). Dror Rawitz was supported in part by the Israel Science Foundation (grant no. 497/14).

of sub-modularity is geometric coverage by sensors. Each additional sensor increases total coverage by the entire area it covers at best, but sometimes less if it overlaps areas already covered by other sensors [10]. For sub-modular functions, greedy selection guarantees a constant factor of approximation and the best possible by any algorithm [11], [12].

Greedy selection depends on pairwise coverage values, at least for selection of the first sensor or location [1]-[3], [9]. Three common ways of aggregating pairwise coverage values are by taking the maximum coverage by any selected sensor (denoted max) [1], [2], the total coverage by all selected sensors (denoted sum)) [13]-[15], and the probability of correct prediction by at least one selected sensor (denoted prob) [4]-[8]. max is based on a pessimistic assumption where coverages overlap, while sum is based on an optimistic assumption where coverages are disjoint. prob is in the middle, and it is based on an independence assumption. These functions can help determine which sensors to select if the final coverage function is unknown or too difficult to calculate The advantage of using them is that they depend on pairwise values of coverage between sensors only, a minimal amount of information for any selection algorithm.

In this paper, we would like to understand what guarantees can be made about coverage by one function when selection is made according to another. This helps determine how useful selection is based on these assumptions, and what can be said in general about selection when the coverage function is unknown or subject to change. We take a graph theoretic approach to the selection problem, modeling sensors as nodes and predictability relationships as weighted directed edges. Since max, sum, and prob are all sub-modular, greedy selection is the most natural algorithm to be applied. We use this to prove bounds on the approximation of selection according to one function when evaluating by another. We prove that in extreme cases, selection by max is $\Theta(\sqrt{n})$ times better than selection by sum and prob when evaluated by max, where n is the number of nodes and each node has a unit value, and likewise in reverse, while selection by prob and sum approximate each other within a constant factor. We apply the greedy algorithm to synthetic and real data sets and show that the error can be as much as 50%.

While the solution found using one function can be very

bad for another, a hybrid algorithm that switches the selection criteria between the max and sum functions works very well. For the three functions max, sum, and prob, it approximates the greedy solution according to the correct function by a constant factor, and in practice reduces the error by 66% and almost entirely in some cases. Since max and sum are the two extremes in sub-modular functions that aggregate edge weight values, it would seem that this approach would be a good approximation for any sub-modular function. Unfortunately, it only has an $\Omega(n^{1/4})$ approximation guarantee, as we prove.

The remainder of the paper is organized as follows. We formally describe the problem in Section III and solutions in Section IV. In Section V we show the price of using the wrong selection criteria, and study the hybrid algorithm in Section VI. Section II is a brief review of related work, and the conclusion and discussion is in Section VII.

II. RELATED WORK

The prob and sum criteria are mostly commonly found in the literature on sensor coverage. Selection under the prob criterion directly corresponds to sensor deployment on a grid [4]-[8]. In this problem, the objective is to deploy sensors to the locations that maximize total coverage of the grid. For every pair of points, there is a probability of a sensor located at one point detecting an event at the other, and coverage is defined by the probability that at least one sensor detects an event. Value fusion is a different way to detect events [13]-[15]. A decision that an event occurred is made if the sum of all signal strengths is above some threshold. This corresponds directly to the sum criterion. Under both coverage models, link values between points are based on the signal strength between points, which means that the sum and prob criteria equally apply to the same link values. The max criteria is more commonly found in the literature on sensor selection. [1], [2] only use pairwise coverage values, just as in the max criteria, under the assumption that only one sensor is used to predict the value of another. Other approaches use more complex aggregate functions of sensor prediction [3], [9], whose coverage values are defined by sub-modular functions. [1], [3], [6]-[8], [10] all use greedy selection under the assumption of sub-modular coverage functions.

The $\frac{e-1}{e}$ approximation factor of maximizing submodular functions by greedy selection was proven by Nemhauser, et al. [12]. This result only applies when unit costs are associated with selecting a location. Hochbaum, et al. [16] give a simpler proof for a specific case of the problem, maximum k-coverage, which is to maximum weighted coverage of objects by k sets. Khuller, et al. [11] extend this result to the general case of budgeted maximum coverage (by non-unit cost sets) using a modified greedy algorithm. They further prove that this is the best possible factor unless P = NP based on a result of Feige, et al. [17]. This result in turn is extended to all submodular functions by Sviridenko [18], effectively extending the result of Nemhauser to instances with non-unit costs.

III. PROBLEM FORMULATION

We model the sensor selection problem as follows. Given is a directed graph G = (V, E) whose nodes have positive weights and costs and whose edges have weights in [0, 1]; also given is a budget k. The task is to choose a subset of nodes of total cost at most k. The coverage value of a node v_i is a function of which nodes are selected; specifically a function of which chosen nodes have edges directed toward v_i and of the weights of those edges. We will consider several such functions. The value of a solution is the weighted sum of all nodes' coverage values. The objective is to maximize this value.

In this model, each node v_i represents a sensor. There are n such nodes. The cost c_i of v_i is the cost of selecting the corresponding sensor, such as its cost of deployment or bandwidth consumption. The weight w_i of v_i is its value in monitoring the network. For example, some sensors monitor more critical areas than others and thus have higher weight. The weight w_{ii} of edge (i,j) is the fraction of v_i 's value that can be predicted by v_i . The aggregate coverage of v_i is the fraction of its value than can be predicted by the selected nodes.

Because all functions will have the property that a zeroweight edge contributes zero to coverage, we assume for notational convenience and without loss of generality that the graph is complete. We also assume that all nodes have weight-1 self-loops, meaning that selecting a node gives it full coverage.

Let S be a given solution and $W_a(S, v_i)$ be the coverage of node v_i by S under coverage function a. The coverage functions are max, sum, and prob. Under max, v_i is predicted by only one node—the one in S with the maximum weighted edge to v_i —so coverage under max is the weight of that edge. Under sum and prob, v_i is predicted collectively by all nodes in S. Under sum, the edge weights represent the fraction of the value of v_i predicted by each node in S, so coverage is the sum total of those edge weights up to the value 1. Under prob, the edge weight w_{ii} represents the probability of correctly predicting the value of v_i by v_i , so coverage is the probability of correct prediction by at least one sensor in S. These are more formally defined as follows:

- $\begin{array}{l} \bullet \ \ W_{max}(S,v_i) = \max_{v_j \in S} w_{ji} \\ \bullet \ \ W_{sum}(S,v_i) = \min\{\sum_{v_j \in S} w_{ji}, 1\} \\ \bullet \ \ W_{prob}(S,v_i) = 1 \Pi_{v_j \in S} (1-w_{ji}) \end{array}$

For a given coverage function a, the value of a selection S, denoted $W_a(S)$, is:

$$W_a(S) = \sum_{v_i \in V} w_i W_a(S, v_i) \tag{1}$$

The objective is to find the set S that maximizes $W_a(S)$ such that $\sum_{v_i \in S} c_i \leq k$. A problem instance I is defined by the graph G = (V, E), including the vertex costs, vertex weights, edge weights, and budget k. For problem instance I, let $ALG_a(I)$ indicate a solution for I obtained by running the adaptive greedy algorithm (described in Section IV) according

```
V \colon \text{list of all sensors } v_i S \leftarrow \{\} \text{for } i \leftarrow 1, k \text{ do} \text{Select } v \in V \text{ that maximizes } W_a(S \cup \{v\}) - W_a(S) S \leftarrow S \cup \{v\} V \leftarrow V - \{v\} \text{end for} \text{return } S
```

Fig. 1. Adaptive greedy selection

to $W_a(\cdot)$, for $a \in \{max, sum, prob\}$, and $W_b(ALG_a(I))$ indicate the value of this solution according to criteria $b \in \{max, sum, prob\}$.

IV. PROBLEM ANALYSIS

Here we analyze the sensor selection problem and solutions. There are four categories of problem instances to consider: when nodes have unit (uniform) costs and weights ($w_i = c_i = 1$ for all i), unit costs and arbitrary weights, arbitrary costs and arbitrary weights, and arbitrary costs and unit weights. Maximizing $W_a(S)$ can be shown to be NP-hard by a simple reduction from the dominating set problem, setting all node weights and costs to one and edge weights to one or zero, or from the knapsack problem, setting all edge weights to zero.

The natural selection algorithm for these coverage functions is adaptive greedy selection because it has the best approximation guarantee of all algorithms, $\frac{e-1}{e}$ [11]. For instances with unit costs, the algorithm selects k nodes one-by-one, each time selecting the node with the maximum increase in coverage by nodes already selected (Fig. 1). For instances with arbitrary costs, the best of two categories of sets is selected: all feasible subsets of one and two nodes, and all feasible sets of three nodes greedily appended by nodes with the maximum increase in coverage density, which is the coverage increase divided by the cost [11], [18]. Note that the asymptotic runtime of greedy selection is $O(n^3)$ when nodes have unit costs and $O(n^6)$ when nodes have arbitrary costs. This can be reduced for W_{max} , but the best known approximation factor in this case is $\frac{\sqrt{e}-1}{\sqrt{e}}$ [11].

Another natural algorithm is to select nodes in descending order of their initial coverage density (i.e., coverage by each node on its own), which we call static greedy selection (denoted STAT) because the nodes are ordered *a priori*. This is a natural choice because it selects nodes by their maximum possible contribution. However, the approximation bounds are far worse than adaptive greedy selection— \sqrt{n} when nodes have unit weights, as proven in the next section.

V. PRICE OF USING THE WRONG OBJECTIVE FUNCTION

Now consider the following scenario. We compute a solution by running ALG_a for some objective function a, but in fact the solution is evaluated based on a second objective b. Since we could not have expected to find an optimal solution according to either objective, we would have liked to at least use the greedy algorithm optimizing for the correct objective

function. In this section, we analyze the error in using the wrong objective function. We limit our study to cases with unit node costs because of the complexity of implementing and executing the greedy solution for arbitrary costs.

We define the price of using the wrong objective function, analogously to Price of Anarchy, as the maximum ratio of the correct values of solutions computed by the two algorithms.

Definition V.1. For any two objective functions a and b, let $d(a,b) = \max_I \frac{W_a(ALG_a(I))}{W_a(ALG_b(I))}$ be the maximum ratio between solutions obtained by the greedy algorithm optimized independently for objectives a and b but evaluated according to objective a.

The "distance function" d(a, b) is the worst case when selecting according to b and evaluating according to a, but it is expressed as how well selecting according to a is for a cleaner presentation. We first study the bounds on d(a, b) for all pairs between max, sum, and prob in Section V-A, and compare static and adaptive greedy selection. What is most noteworthy is that the distance between max and either sum or prob is in the order of $O(\sqrt{n})$ when nodes have unit weights, which is just as bad as approximation of the optimal solution by static greedy selection. Because the cases used to prove the bounds on d(a,b) are pathological, meaning that they do not necessarily represent common cases, in Section V-B we study the ratios on randomly generated graphs. Here, too, the error is large, in some cases up to 50% and more as the size of the graph increases. In Section V-C, we go one step further and show that even in graphs whose edge weights are set according to real data sets, there is a large gap in the total value when nodes are selected by the wrong criteria.

A. Worst-Case Analysis

In this section, we prove upper and lower bounds on d(a,b) for all pairs $(a,b) \in \{max, sum, prob\}^2$. An upper bound is proven by showing that it holds for all instances, while a lower bound is proven by providing an infinite family of instances, for all values of n, that satisfy it. To simplify the constructions and arguments for the lower bounds, we assume the existence of an adversary who decides how to break ties when several nodes have the same marginal value. It can be shown how to modify these instances to accommodate any tie breaking rule. The loss in the bounds will be $\Theta(\frac{1}{k})$.

We first prove upper and lower bounds on d(a,b) for graphs with unit node weights, then for graphs with arbitrary node weights, and finally on static greedy selection. Note again that our study is limited to graphs with unit node costs.

Unit node weights

The bounds on d(a,b) for graphs with unit node weights are not tight, but they are tight up to a constant. The general observation is that the relationship between max and either sum or prob is $\Theta(\sqrt{n})$, while the relationship between sum and prob is $\Theta(1)$, and that these relationships are nearly symmetric. Later we show that this is not so when nodes have arbitrary values.

The following is a universal upper bound for all pairs of objectives for graphs with unit costs and weights.

Theorem V.1. When nodes have unit weights and for all ordered pairs $(a,b) \in \{max, sum, prob\}^2, d(a,b) \leq \sqrt{n}$.

Proof: First, the value of the optimal solution is $\leq n$ and the value of any solution with budget k is at least k, so $d(a,b) \leq \frac{n}{k}$. Second, the same node is selected first according to all three objectives, so its value A is necessarily the same for all three greedy algorithms. Since the best the "right" greedy algorithm could possibly do is get that value k-1 more times for a total of Ak, and the worst the "wrong" greedy algorithm could do is get nothing else at all, for a total of just A, $d(a,b) \leq k$. Since $\frac{n}{k}$ is a decreasing function of k and k is an increasing function of k, the lower bound of the worst case of these bounds is when these two equal each other (when $\frac{n}{k} = k$), meaning $k = \sqrt{n}$.

The upper bound can be strengthened for d(sum, prob) and d(prob, sum).

Theorem V.2. When nodes have unit weights, $d(sum, prob), d(prob, sum) \le \left(\frac{e}{e-1}\right)^2$

Proof: Let $z = \frac{e}{e-1}$. For every subset $R \subseteq V$,

$$W_{prob}(R) \le W_{sum}(R) \tag{2}$$

and

$$W_{sum}(R) \le z \cdot W_{prob}(R) \tag{3}$$

To prove (2), note that for any node v_i ,

$$1 - \prod_{v_j \in R} (1 - w_{ji}) = \sum_{v_j \in R} \left(\prod_{v_{k < j} \in R} (1 - w_{ki}) \right) w_{ji} \le \sum_i w_{ji}$$
(4)

and

$$1 - \prod_{j} (1 - w_{ji}) \le 1 \tag{5}$$

To prove (3), let $y = \sum_{v_j \in R} w_{ji}$ for any node v_i and note that

$$\prod_{v_j \in R} (1 - w_{ji}) \le \left(\frac{\sum_{v_j \in R} (1 - w_{ji})}{n}\right)^n$$

$$= \left(\frac{n - y}{n}\right)^n = \left(1 - \frac{y}{n}\right)^n \le e^{-y} \quad (6)$$

The first inequality is a consequence of the arithmetic-geometric mean inequality, and the last inequality can be proven by setting $a=\frac{y}{n}$ and noting that $1-a\leq e^{-a}$ when $a\leq 1$. From the above inequality and the definition of W_{sum} , we have

$$\frac{W_{sum}(R)}{W_{prob}(R)} \le \frac{\min[y, 1]}{1 - e^{-y}} \tag{7}$$

Taking the derivative of (7) shows that it increases with y when $y \le 1$ and decreases with y when $y \ge 1$. Therefore, the maximum value of (7) is $\frac{e}{e-1}$, when y=1.

Now let S be the set found by ALG_{sum} , P be the set found by ALG_{prob} , S' be the optimal set under sum, and P' be the

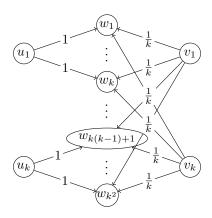


Fig. 2. Instance satisfying $\frac{1}{2}\sqrt{n}$ lower bound on d(max,sum) and d(max,prob) when nodes have unit weights

optimal set under prob. By the above inequalities and by the guaranteed approximations of ALG_{prob} and ALG_{sum} we get:

$$W_{sum}(S) \le z \cdot W_{prob}(S)$$
 by (3)

$$\leq z \cdot W_{prob}(P')$$
 optimality of P' (9)

$$\leq z^2 \cdot W_{prob}(P)$$
 approximation guarantee (10)

$$\leq z^2 \cdot W_{sum}(P) \quad \text{by (2)} \tag{11}$$

implying

$$d(sum, prob) = W_{sum}(S)/W_{sum}(P) \le z^2$$
 (12)

and

$$W_{prob}(P) \le W_{sum}(P) \qquad \text{by (2)} \tag{13}$$

$$\leq W_{sum}(S')$$
 optimality of S' (14)

$$\leq z \cdot W_{sum}(S)$$
 approximation guarantee (15)

$$\leq z^2 \cdot W_{prob}(S) \quad \text{by (3)} \tag{16}$$

implying

$$d(prob, sum) = W_{prob}(P)/W_{prob}(S) \le z^2 \tag{17}$$

Next we describe the instances satisfying lower bounds on all these pairs.

Theorem V.3. When nodes have unit weights, $d(max, sum) \ge \frac{1}{2}\sqrt{n}$.

Theorem V.4. When nodes have unit weights, $d(max, prob) \ge \frac{1}{2}\sqrt{n}$.

Proof: Create three sets of nodes, $\{u_1 \dots u_k\}$, $\{v_1 \dots v_k\}$, and $\{w_1 \dots w_{k^2}\}$, for a total of $n=k^2+2k$ nodes. From each u_i , create a 1-weighted edge to every $w_{(i-1)k+1} \dots w_{(i-1)k+k}$, and from each v_i , create a $\frac{1}{k}$ -weighted edge to every w_j . See Fig. 2. Both u_1 and v_1 have coverage value k+1 when selected first, so the adversary forces ALG_{max} to select u_1 first and ALG_{sum} and ALG_{prob} to select v_1 first. ALG_{max} then selects all remaining u_i , and ALG_{sum} selects all remaining v_i . Under W_{prob} , both u_i and v_i increase coverage by $(1-1/k)^{i-1}k+1$ when $v_1 \dots v_{i-1}$ are selected first, so the

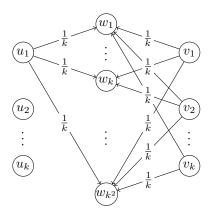


Fig. 3. Instance satisfying $\frac{1}{2}\sqrt{n}$ lower bound on d(sum,max) and $\frac{e-1}{e}\frac{1}{2}\sqrt{n}$ lower bound on d(prob,max) when nodes have unit weights

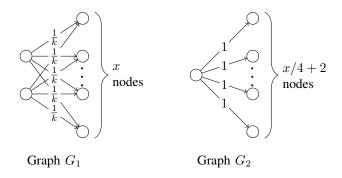


Fig. 4. Instance satisfying 4/3 lower bound on d(sum, prob) when nodes have unit weights

adversary forces ALG_{prob} to keep selecting from v_i . The total value of ALG_{max} will be $k(k+1)=k^2+k$, and the total value of ALG_{sum} and ALG_{prob} will be k+k=2k. Since $k+1=\sqrt{n+1}$, the ratio between them will be $\frac{k+1}{2}=\frac{\sqrt{n+1}}{2}$, which is greater than $\frac{\sqrt{n}}{2}$.

Theorem V.5. When nodes have unit weights, $d(sum, max) \ge \frac{1}{2}\sqrt{n}$.

Proof: Create three sets of nodes, $\{u_1 \dots u_k\}$, $\{v_1 \dots v_k\}$, and $\{w_1 \dots w_{k^2}\}$, for a total of $n=k^2+2k$ nodes. From u_1 create a $\frac{1}{k}$ -weighted edge to every w_j , and from each v_i , create a $\frac{1}{k}$ -weighted edge to every w_j . See Fig. 3. The adversary forces ALG_{sum} to select all v_i and ALG_{max} to select all u_i . Therefore, the total value of ALG_{sum} will be k^2+k , while the total value of ALG_{max} will be 2k. The ratio between them will be $\frac{k+1}{2}$, or $\frac{\sqrt{n+1}}{2}$, which is greater than $\frac{\sqrt{n}}{2}$.

Theorem V.6. When nodes have unit weights, $d(sum, prob) \ge 4/3$.

Proof: Create the two graphs in Fig. 4:

- a complete bipartite graph G_1 between k=2 nodes and x nodes, with all edge weights equal to 1/2
- a star graph G_2 with x/4+2 nodes, with all edge weights equal to 1

Then ALG_{sum} will choose both left nodes of G_1 for a total

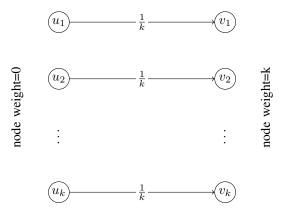


Fig. 5. Instance satisfying $\frac{1}{2}n$ lower bound on d(max, sum) and d(max, prob) when nodes have arbitrary weights

value of x + 2, whereas ALG_{prob} will choose one left node of G_1 and the center node of G_2 for a total value of $\frac{3}{4}x + 3$.

Theorem V.7. When nodes have unit weights, $d(prob, max) \ge \frac{e-1}{e} \frac{1}{2} \sqrt{n}$.

Proof: Create the graph in Fig. 3. ALG_{prob} will select $v_1 \dots v_k$, with total value $k + (1 - (1 - \frac{1}{k})^k)k^2$, while the adversary forces ALG_{max} to select $u_1 \dots u_k$, with total value 2k. The ratio between them as k approaches ∞ is $1 + \frac{e-1}{e} \frac{\sqrt{n+1}-1}{2}$, which is greater than $\frac{e-1}{e} \frac{\sqrt{n}}{2}$.

Theorem V.8. When nodes have unit weights, $d(prob, sum) \ge \frac{e}{e-1}$.

Proof: Create the graph in Fig. 2. The adversary forces ALG_{prob} to select all u_i with a total value of $k^2 + k$, and ALG_{sum} to select all v_i with a total value of $k + (1 - (1 - \frac{1}{k})^k)k^2$.

Arbitrary node weights

When the nodes have arbitrary values, the bounds on d(max, sum) and d(max, prob) are higher, as follows.

Theorem V.9. When nodes have arbitrary weights, $d(max, sum) \ge \frac{1}{2}n$

Proof: Create two sets of sets of nodes, $\{v_1 \dots v_k\}$, all weighted 0, and $\{u_1 \dots u_k\}$, all weighted k, for a total of n=2k nodes. From each v_i , create a $\frac{1}{k}$ -weighted edge to every u_i . See Fig. 5. ALG_{max} selects all u_i with a value of k, but the adversary forces ALG_{sum} to select all v_i with the coverage value of k under the sum metric. ALG_{max} will have total coverage $k \cdot k = k^2$, while ALG_{sum} will only have total coverage of k. The ratio between them is $\frac{k^2}{k} = k$, or $\frac{n}{2}$.

Theorem V.10. When nodes have arbitrary weights, $d(max, prob) \ge \frac{1}{2}n$

Proof: Create the same graph as in Theorem V.9. Note that under W_{prob} , selecting v_i or u_i increases coverage by $(\frac{k-1}{k})^{(i-1)}k$. The adversary forces ALG_{prob} to select $v_1 \dots v_k$ for total coverage k while ALG_{prob} selects from $u_1 \dots u_k$ with

the total coverage of k^2 . The ratio between them is k, or $\frac{n}{2}$.

However, the reverse–d(sum, max) and d(prob, max)–appears not to change, i.e., to remain bounded by $O(\sqrt{n})$. Also, d(sum, prob) and d(prob, sum) seem to not much change. Intuitively, the reason seems to be that even absent unit values the n/k bound still holds in the cases of d(sum, max) and d(prob, max) because the error is conservative, and so (intuitively) ALG_{max} can anyway choose the k highest-value nodes, in which case the worst case for that bound is actually when values are unit.

Static greedy selection

An alternative comparison is between adaptive selection and static selection, by analogy to online and offline algorithms. Note that when v_i is covered by only one node, its coverage is the same under all functions— w_{ji} —so the static greedy algorithm provides the same solution for all three functions. Nonetheless, we compare the value of the static algorithm's result, evaluated under a given objective, compared to the value of the corresponding dynamic algorithm, as above.

Note that the same node is selected first by the static and dynamic algorithm, regardless of the objective function, so $\frac{W_a(ALG_a(I))}{W_a(STAT)} \leq \sqrt{n}$ according to Theorem V.1. In the following theorem, we prove that this is the lower bound as well.

Theorem V.11. When nodes have unit weights,
$$\max_I \frac{W_a(ALG_a(I))}{W_a(STAT)} \ge \sqrt{n}$$

Proof: Create a complete graph with k nodes and k-1 star graphs with k nodes. Set all edges weights to 1. The total number of nodes is $k+k\cdot(k-1)=k^2$. ALG_a will select one of the nodes of the complete graph with a coverage value of k and then select the center of each star graph with the coverage value k for each center. The adversary forces ALG_{stat} to select only from the complete graph for a total value of k, while ALG_a will have a total value of $k+k\cdot(k-1)=k^2$. The ratio between them is k, which equals \sqrt{n} .

A peculiar situation arises when ALG_{max} and STAT are applied, but the result is evaluated according the W_{sum} . For the graph in the proof of Theorem V.5, STAT will make the same selection as ALG_{sum} , thus the same ratio applies. Formally stated, $\max_I \frac{W_{sum}(STAT)}{W_{sum}(ALG_{max}(I))} \geq \frac{\sqrt{n}}{2}$. So we see that the static algorithm is sometimes better than the dynamic algorithm, albeit when both are applied to the wrong objective function.

B. Randomly Generated Graphs

We study the error in selecting by the wrong criteria for graphs with randomly assigned node and edge values. This allows us to study the factors that contribute to the error in a controlled manner. We tried the four network topologies listed in Table I. The bipartite topology was chosen to reflect the topologies in the examples in Section V. The node and edge weights were both drawn from either the uniform distribution or the power law distribution as defined in Table II.

TABLE I NETWORK TOPOLOGIES USED IN RANDOM SIMULATIONS

name	description
complete sparse X triangle bipartite	all edges edge (i,j) exists with probability X edge (i,j) exists when $i \leq j$ edge (i,j) exists when $i \leq \sqrt{n}$ and $j \geq \sqrt{n}$

TABLE II
PROBABILITY DISTRIBUTIONS OF NODE AND EDGE WEIGHTS

distribution	parameters
uniform	$f(x) = \frac{1}{b-a}x$
	for w_i : $a \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}, b = 1$
	for w_{ij} : $a = 0, b \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$
power law	$f(x) = \frac{\alpha + 1}{b^{\alpha + 1} - a^{\alpha + 1}} x^{-\alpha}$
	$\alpha \in \{0, 1, 2, 3, 4\}$
	for w_i : $a = 0, b = 1$
	for w_{ij} : $a = 0, b = 0.2$

The complete and sparse topologies and uniform distribution are obvious starting points, while the triangle and bipartite topologies and power law distribution highlight some of the extremes in error. All node costs were set to one and the networks contained 100 nodes or more.

For each set of parameters, we created up to 20 random instances, derived the adaptive greedy solution according to each criteria $a \in \{max, sum, prob\}$, and calculated the ratio $\frac{W_a(ALG_b)}{W_b(ALG_b)}$, denoted r(a,b), for all pairs $(a,b) \in \{max, sum, prob\}^2$. This ratio is a measure of how bad selection by a is with respect to criteria b, which is done by comparing its value under criteria b to the value of selection according to the "right" criteria b. Below we discuss the average of all instances for a specific set of parameters and budget value.

We first look at sum and prob evaluated by max. Fig. 6(b) shows the average ratios r(sum, max) and r(prob, max) for each budget value for the complete graph using uniform distributions, with node weights in the range [0, 1], edge weights in the range [0, 0.2], and n = 500. r(sum, max) is as low as 0.72 and r(prob, max) is as low as 0.82. As x or y increase, these values increase and r(prob, max) approaches r(sum, max) (Table III). Fig. 6(a) is for the same parameters in Fig. 6(b), except that n = 100. While r(sum, max)is not much different, r(prob, max) is much further from r(sum, max), which is the case over all combinations of x and y. This suggests that r(prob, max) converges to r(sum, max)as n increases. Under the power law distribution, not only does r(sum, max) decrease as α increases for either the edge or node weight distributions, but it decreases as n increases as well. Similar results were found for the other topologies. Sparseness was not a contributing factor to the ratios.

When evaluating by the other two criteria, sum and prob, the error of using the wrong criteria was minimal in the com-

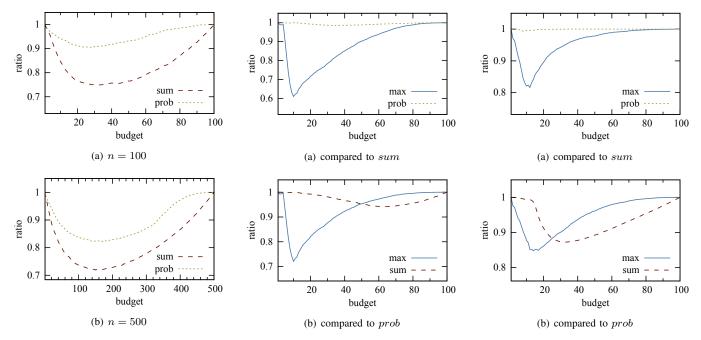


Fig. 6. Comparison to max for a complete graph, uniform distribution, node range [0,1], and edge range [0,0.2]

Fig. 7. Comparisons for a bipartite graph, uniform distribution, node range [0,1], edge range [0,0.2], and n=100

Fig. 8. Comparisons for a triangle graph, uniform distribution, node range [0,1], edge range [0,0.2], and n=100

TABLE III Worse approximation of ALG_{max} for a complete graph, uniform distribution, n=500, and different values of node weight parameter a (along rows) and edge weight parameter b

(ALONG COLUMNS)

	approximation by ALG_{sum}							
	0.2	0.4	0.6	0.8	1.0			
0	0.719	0.815	0.886	0.947	0.973			
0.2	0.796	0.872	0.922	0.964	0.973			
0.4	0.863	0.914	0.949	0.977	0.974			
0.6	0.912	0.946	0.97	0.977	0.974			
0.8	0.96	0.976	0.98	0.978	0.975			
1.0	0.985	0.984	0.981	0.975	0.975			
	approximation by ALG_{prob}							
	0.2	0.4	0.6	0.8	1.0			
0	0.822	0.838	0.887	0.946	0.989			
0.2	0.844	0.884	0.923	0.964	0.99			
0.4	0.879	0.918	0.95	0.977	0.989			
0.6	0.92	0.95	0.97	0.986	0.991			
0.8	0.962	0.976	0.986	0.99	0.99			
1.0	0.987	0.989	0.991	0.99	0.99			

plete and sparse graphs, with r(sum, max) and r(prob, max) being above 0.95 in most cases. In the bipartite (Fig. 7) and triangle (Fig. 8) graph with uniform distribution and node weights in [0,1], there was a significant error when selecting by max and evaluating by either criteria, with r(max, sum) as low as 0.6 when edge weights are in [0,0.2]. While there was minimal error for r(prob, sum) and r(sum, prob) in the bipartite graph, we see a nice dichotomy in the triangle graph.

r(prob, sum) is minimal, but r(sum, prob) is nearly as low as r(max, sum). In both graphs, r(sum, prob) became lower than r(max, sum) as x and y increase.

C. Real Data Sets

Since the study of randomly generated graphs leaves us with a question if these errors occur in reality, we also studied the error in graphs based on real data sets. We tried many publicly available data sets [19]–[21] and present the six most significant results. We focus on the gap between max and sum and exclude prob from the evaluation. Even here, selecting by the wrong criteria is as low as 50%. What was interesting is that when the gap is large under one criteria, it is small under the other. In each case, a different approach was used to set the edge weights, showing that these results are independent of how the edges weights are set.

Intel Berkeley Research lab [3], [19]. We used the first 30,000 epochs from 48 out of 54 sensors (motes 5, 15, 16, 17, 18, and 28 were excluded) to calculate regression coefficients between pairs of data streams for both light and humidity data. We linearly interpolated the missing data and normalized the data by replacing each sample in a stream with the number of standard deviations it falls from the mean for that stream. We set the edge weights to 1 minus the mean-squared error of the predicted data along each link. In Fig. 9, we can see for both that coverage under max of the selection by sum is about 10% to 15% less than that of the selection by max.

EPANET Water Data [22]. We used EPANET 2.0 [22] to simulate water flow in a pipe network with 125 junctions between various pipes [23]. The chlorine concentrations in the sensors were used for the prediction process. We normalized

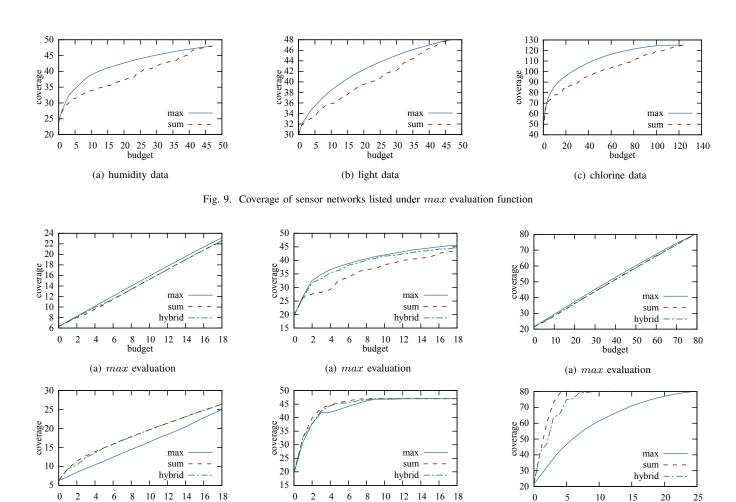


Fig. 10. Coverage of EIES network under specified evaluation function

(b) sum evaluation

Fig. 11. Coverage of intra-organisational network by specified evaluation function

(b) sum evaluation

budget

Fig. 12. Coverage of SPECT data by specified evaluation function

(b) sum evaluation

budget

the data and set the link weights as we did for the Intel Berkeley data, and had similar results (Fig. 9).

EIES networks [24], [25]. This network was constructed from a matrix with the number of messages sent among 32 researchers. This network represents how much information we can get from this operation. Since the maximum number of messages on one edge is 559, we divided the number of messages on each edge by 600. In Fig. 10, we see a gap as large as 26% when evaluating by sum but a small gap when evaluating by max.

Intra-organisational networks [26], [27]. This is a network of 46 employees in a consulting company that rank each other on a scale from 0 to 5 based on how often they turned to them for work-related information or advice. We used this dataset as a relationship graph between all employees, setting the edge weights to the ranks divided by 5. The idea would be to select the set of employees that have received the most information from other employees. In Fig. 11, we see a gap as large as 20% when evaluating by max but a small gap when evaluating by sum, as opposed to the EIES network.

SPECT Heart [21]. The dataset describes cardiac Single Proton Emission Computed Tomography (SPECT) images. The database of 267 SPECT image sets (patients) was processed to obtain 22 binary feature patterns. We used 80 instances to test the selection algorithms. The goal is to select a subset of images that gives as much information of the entire dataset, so a researcher only needs to study those subset of images instead of studying the entire image dataset. We set the edge weights according to the Kendall Tau rank distance between the binary feature vectors of each pair of images. In Fig. 12, we see a gap as large as 50% when evaluating by sum but a small gap when evaluating by max. The experiment shows under the sum evaluation using adaptive sum algorithms we need only 6 images to predict the whole network but under the adaptive max algorithm we need 25 images (Changing the metric from sum to max results in 80 for adaptive sum and 80 for adaptive max algorithm).

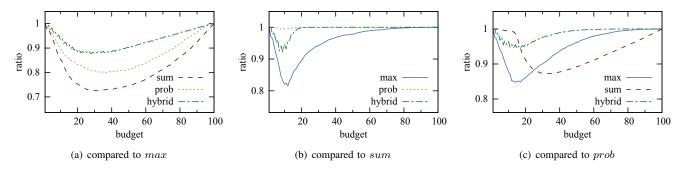


Fig. 13. Coverage of the triangle graph using hybrid selection

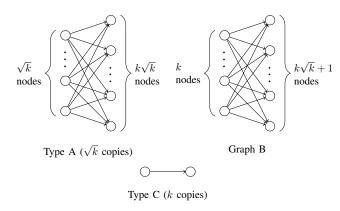


Fig. 14. Instance proving $n^{1/4}$ lower bound approximation ratio of max-sum hybrid selection of arbitrary sub-modular functions

VI. HYBRID GREEDY SELECTION

We saw in the previous section that the approximation error in the worst case can be $O(\sqrt{n})$ or even more when nodes have arbitrary weights, and even in practice can be large. One way to avoid this problem is to use a hybrid of the criteria. In the hybrid solution, the criteria by which nodes are selected is switched between max and sum round-robin. That is, the second node selected is the one that increases the value according to sum, the third according to max, the fourth back to sum, and so on. It can be easily proved that this gives a $\frac{1}{2}$ -approximation of the greedy solution according to the right criteria out of max and sum. Since the selection is greedy and half the budget is dedicated to each criteria, then at least half of the solution is approximated.

In practice, the hybrid solution is very good. For the real data sets in Section V-C, the value of the hybrid solution is remarkably close to the value of solution by the right criteria (Figures 10, 11, and 12). Fig. 13 revisits the triangle graph of Fig. 8 with coverage by the hybrid selection added. It shows that it reduces the gap between max and the other criteria by up to 2/3. This trend was seen throughout all the simulations.

The Wild Conjecture and Why it's Wrong

One motivation for the analysis in this paper is to see how well we do if we optimize for one objective and then it unfortunately turns out that the solution is actually evaluated based on another objective. Since the greedy solution itself gives an $\frac{e-1}{e}$ -approximation of the optimal solution, we obtain that if we optimize for a and evaluate according to b, then the solution satisfies an approximation guarantee of $\frac{1}{d(a,b)} \cdot \frac{e-1}{e}$. Since hybrid selection is a $\frac{1}{2}$ -approximation of selection by the right criteria, it is also a $\frac{1}{2} \cdot \frac{e-1}{e}$ -approximation of the optimal solution. Now, max and sum are the two extremes in submodular functions that aggregate edge weight values. We therefore posed the wild conjecture the hybrid solution is even a $\frac{1}{2} \cdot \frac{e-1}{e}$ -approximation of the optimal solution for any submodular function. It turns out that this is wrong. Even worse, the approximation ratio is $\Omega(n^{1/4})$, as we prove in the following theorem.

Theorem VI.1. No hybrid algorithm that greedily selects half the nodes according to the max criteria and the other half according to the sum criteria can guarantee approximation of the optimal solution for an arbitrary sub-modular function by a factor less than $n^{1/4}$.

Proof: Construct \sqrt{k} graphs (denoted A graphs) with \sqrt{k} nodes pointing to $k\sqrt{k}$ other nodes. Create another graph (denoted Graph B) with k nodes pointing to $k\sqrt{k}+1$ other nodes, and finally another graph (denoted Graph C) with k nodes pointing to k other nodes. All edge weights are set to 1/k. The total number of nodes is in the order of k^2 . The budget is k, and the evaluation function is $f(v) = \min\{sum, \frac{1}{\sqrt{k}}\}$. ALG_{max} will select one pointing node from each graph of Type A, one pointing node from Graph B, and $k - \sqrt{k-1}$ pointing nodes from Graph C, while sum will choose all pointing nodes from Graph B. The total coverage under f(v)in both cases is in the order of $\Theta(k)$, so coverage by hybrid selection is also in the order of $\Theta(k)$. The optimal solution is to choose all pointing nodes in all Type A graphs, with coverage in the order of $\Theta(k\sqrt{k})$, so the approximation ratio is $\sqrt{k} = n^{1/4}$.

VII. CONCLUSION

We presented the problem of selecting sensors for prediction by the wrong evaluation criteria. We proved the worst-case approximation ratios when the nodes have uniform cost, showing that the ratio is as high as $\Theta(\sqrt{n})$, when nodes have unit costs and weights and either the selection or evaluation criteria is

max, and even higher when nodes have arbitrary weights. We showed that this is a problem in practice and not just theory by running the algorithms on randomly generated graphs and graphs that represent the relationships in real data sets. In something as basic as a complete graph with random node and edge weights, coverage when selecting by the wrong criteria is up to 35% less than when selecting by the right criteria. We then showed that splitting the budget between criteria in a hybrid greedy selection algorithm does very well in practice. In theory, it guarantees a $\frac{1}{2}$ -approximation of selection by the right criteria, but in practice coverage error was less than half than selection by one of the "wrong" criteria and in one case as good as selection by the "right" criteria. However, this cannot be used as a $\frac{1}{2}\frac{e-1}{e}$ -approximation of the optimal solution for general evaluation functions, as proven.

Some of these bounds need to be tightened, and the more challenging case of arbitrary node costs still needs to be analyzed. We can analyze the problem differently by removing the assumption of self-loops (full coverage of a node by itself). The hybrid algorithm can be studied further. One problem to begin with is how to divide the budget between criteria. Some alternatives to round-robin selection, used in this paper, are to first select half the nodes by one criteria and then the other half by the other, or to randomly choose between criteria. The questions again are how these techniques perform in theory and practice, and if there is any practical difference at all. The interesting question is how this can be used as an approximation algorithm for the optimal solution when the evaluation criteria is not known. Is it possible to guarantee a good approximation factor by splitting the budget between several known criteria?

ACKNOWLEDGMENT

The authors would like to thank Sven Kohler for his extensive assistance with the figures and layout.

REFERENCES

- C. C. Aggarwal, A. Bar-Noy, and S. Shamoun, "On sensor selection in linked information networks," in *DCOSS*, 2011, pp. 1–8.
- [2] A. Das and D. Kempe, "Sensor selection for minimizing worst-case prediction error," in *IPSN*, 2008, pp. 97–108.
- [3] A. Krause, A. P. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.
- [4] J. Zhang, T. Yan, and S. H. Son, "Deployment strategies for differentiated detection in wireless sensor networks," in *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2006, September 25-28, 2006, Reston, VA, USA.* IEEE, 2006, pp. 316–325. [Online]. Available: http://dx.doi.org/10.1109/SAHCN.2006.288436
- [5] N. Ahmed, S. S. Kanhere, and S. Jha, "Probabilistic coverage in wireless sensor networks," in 30th Annual IEEE Conference on Local Computer Networks (LCN 2005), 15-17 November 2005, Sydney, Australia, Proceedings. IEEE Computer Society, 2005, pp. 672–681. [Online]. Available: http://dx.doi.org/10.1109/LCN.2005.109
- [6] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," ACM Trans. Embedded Comput. Syst., vol. 3, no. 1, pp. 61–91, 2004.
- [7] ——, "Uncertainty-aware and coverage-oriented deployment for sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 788–798, 2004.

- [8] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in 2003 IEEE Wireless Communications and Networking, WCNC 2003, New Orleans, LA, USA, 16-20 March, 2003. IEEE, 2003, pp. 1609–1614. [Online]. Available: http://dx.doi.org/10.1109/WCNC.2003.1200627
- [9] F. Bian, D. Kempe, and R. Govindan, "Utility based sensor selection," in Proceedings of the Fifth International Conference on Information Processing in Sensor Networks, IPSN 2006, Nashville, Tennessee, USA, April 19-21, 2006, J. A. Stankovic, P. B. Gibbons, S. B. Wicker, and J. A. Paradiso, Eds. ACM, 2006, pp. 11–18. [Online]. Available: http://doi.acm.org/10.1145/1127777.1127783
- [10] Y. Yang, I.-H. Hou, J. C. Hou, M. Shankar, and N. S. V. Rao, "Sensor placement for detecting propagative sources in populated environments," in *INFOCOM*. IEEE, 2009, pp. 1206–1214.
- [11] S. Khuller, A. Moss, and J. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, no. 1, pp. 39–45, 1999.
- [12] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Mathe-matical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
- [13] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C. Yi, "Data fusion improves the coverage of wireless sensor networks," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, MOBICOM 2009, Beijing, China, September 20-25, 2009*, K. G. Shin, Y. Zhang, R. Bagrodia, and R. Govindan, Eds. ACM, 2009, pp. 157–168. [Online]. Available: http://doi.acm.org/10.1145/1614320.1614338
- [14] E. Amaldi, A. Capone, M. Cesana, and I. Filippini, "Coverage planning of wireless sensors for mobile target detection," in IEEE 5th International Conference on Mobile Adhoc and Sensor Systems, MASS 2008, 29 September - 2 October 2008, Atlanta, Georgia, USA. IEEE, 2008, pp. 48–57. [Online]. Available: http://dx.doi.org/10.1109/MAHSS.2008.4660029
- [15] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for detection of targets traversing a region," *MONET*, vol. 8, no. 4, pp. 453–461, 2003.
- [16] D. S. Hochbaum and A. Pathria, "Analysis of the greedy approach in problems of maximum k-coverage," *Naval Research Logistics*, vol. 45, no. 6, pp. 615–627, 1998.
- [17] U. Feige, "A threshold of In for approximating set cover," J. ACM, vol. 45, no. 4, pp. 634–652, 1998.
- [18] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Oper. Res. Lett.*, vol. 32, no. 1, pp. 41–43, 2004.
- [19] S. Madden, "Intel lab data," http://db.csail.mit.edu/labdata/labdata.html.
- [20] "Tore opsahl datasets," http://opsahl.co.uk/tnet/datasets/.
- [21] K. Bache and M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml/datasets/SPECTF+Heart, 2013.
- [22] "Epanet," http://www.epa.gov/nrmrl/wswrd/dw/epanet.html.
- [23] A. Ostfeld(et.al.), "The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 556–568, 2008. [Online]. Available: http://link.aip.org/link/?QWR/134/556/1
- [24] S. C. Freeman and L. C. Freeman, The networkers network: a study of the impact of a new communications medium on sociometric structure. School of Social Sciences University of Calif., 1979.
- [25] "Eies dataset," http://opsahl.co.uk/tnet/datasets/Freemans_EIES-3_n32. txt.
- [26] R. Cross, A. Parker, C. M. Christensen, S. D. Anthony, and E. A. Roth, The hidden power of social networks. Audio-Tech Business Book Summaries, Incorporated, 2004.
- [27] "Intra-organisational network dataset," http://opsahl.co.uk/tnet/datasets/ Cross_Parker-Consulting_info.txt.