Finding the Needles in the Haystack: Efficient Intelligence Processing

Nedialko B. Dimitrov
Graduate Program in Operations Research and Industrial Engineering
The University of Texas at Austin, Austin, TX 78712
ned@austin.utexas.edu

Moshe Kress
Operations Research Department
Naval Postgraduate School, Monterey, CA 93943
mkress@nps.edu

Yuval Nevo
Operations Research Department
Naval Postgraduate School, Monterey, CA 93943
ynevo@nps.edu

As a result of communication technologies, the main intelligence challenge has shifted from collecting data to efficiently processing it so that relevant, and only relevant, information is passed on to intelligence analysts. We consider intelligence data intercepted on a social communication network. The social network includes both adversaries (e.g. terrorists) and benign participants. We propose a methodology for efficiently searching for relevant messages among the intercepted communications. Besides addressing a real and urgent problem that has attracted little attention in the open literature thus far, the main contributions of this paper are two-fold. First, we develop a novel knowledge accumulation model for intelligence processors, which addresses both the nodes of the social network (the participants) and its edges (the communications). Second, we propose efficient prioritization algorithms that utilize the processor's accumulated knowledge. Our approach is based on methods from graphical models, social networks, random fields, Bayesian learning, and exploration/exploitation algorithms.

Key words: intelligence processing; knowledge; exploration; exploitation

1. Introduction

Effective military and law-enforcement operations depend on reliable, relevant and timely intelligence. It has been postulated that many terrorist events could be avoided, or at least mitigated, if available intelligence was better processed, analyzed and disseminated on time (Gorman 2002, Radack 2011). The advent of sensing technologies – from electro-optical devices to cyber interceptors – has resulted in a plethora of sensors that collect and transmit an unprecedented glut of

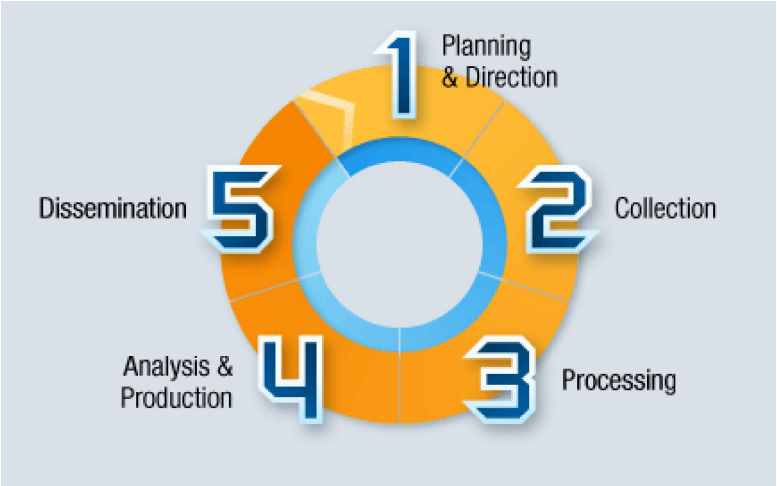


Figure 1 **The intelligence cycle. The cycle consists of the five stages listed in the figure. In the Processing stage, collected data is sorted into relevant and irrelevant data. Only the relevant data should be passed to the Analysis and Production stage to maximize effectiveness of the analysts. We focus on this critical Processing stage, where a glut of data that could overwhelm processors.**

data. These data need to be processed and analyzed in a timely manner in order to produce useful information for operations.

Intelligence operations are typically depicted as a cycle (Kaplan 2012) comprising five stages, as shown in Figure 1. In the *Planning and Direction* stage the intelligence product consumer determines the required information and specifies the queries for which concrete answers are sought. Intelligence data are collected by various sensors — human, visual, electronic, communication, geospatial — in the *Collection* stage. During *Processing*, the raw data obtained from the sensors, which range from observations, to verbal and noisy messages, need to be processed and filtered in order to create an effective analysis. In particular, data are screened such that relevant, and only relevant, intelligence items are passed on to the *Analysis and Production* stage. The objective of the *Analysis and Production* stage is to gain insights from the processed data, and to confirm or reject certain hypotheses. The final stage, *Dissemination*, generates reports, presentations and other communications that deliver the final product of the intelligence analysis to its consumers.

Stage three, *Processing*, is critical. Many intelligence queries are time-sensitive; relevant information needs to be passed on for analysis as quickly as possible to respond to eminent threats and contingencies. The main challenge is to quickly screen the data and provide analysts with a set of relevant, and only relevant, items within a given time window. In this paper, we focus on that stage — *Processing* — and propose efficient screening policies for processing raw intelligence data collected from a social network.

The operations research literature on intelligence operations is quite limited (Kaplan 2012). The first model addressing situational awareness is Deitchman's Guerrilla model (Deitchman 1962),

which was followed by Schaffer (1968). These models capture information asymmetry between a regular, exposed, force and a guerrilla unit that blends in the environment. In the early 60's the CIA utilized Bayes' Rule for assessing the situation during the Cuba missile crisis (Zlotnik 1967). That research had been classified as SECRET until 1994. More recent related papers analyze the management of secrets (Steele 1989), model the development of secret projects (Harney et al. 2006, Godfrey et al. 2007), and estimate the number of secret terror plots in progress (Kaplan 2010).

In this paper we consider an adversarial (e.g. terrorist) social communication network (e.g. phone calls, e-mails) that is subject to interception by a friendly intelligence agency. The nodes of the network correspond to communicators, and the edges represent communications between pair of nodes. Both adversaries and benign nodes may be captured in the interception of communications. We propose a methodology for efficiently answering the following question: *Given a set of intercepted communications how should an intelligence processor prioritize the screening of these communications such that the expected number of relevant intelligence items that are passed on to the analysts is maximized*?

To the best of our knowledge, this question has not been addressed in the open literature. Moreover, our approach for answering this question – addressing both the communications (edges) and the communicators (nodes) in a unified way is novel, and can be extended in many ways. Utilizing methods from graphical models, social networks, random fields, Bayesian learning, and exploration/exploitation algorithms, we develop a model for the processor's accumulated knowledge regarding the value of the nodes as information providers, and the relevance potential of communications on the edges. We formulate an optimization problem for selecting most potentially relevant communications, and describe an exact method for solving it. We also develop and test several heuristic methods and obtain some interesting insights regarding the screening process.

The rest of the paper is organized as follows. Section 2 describes the basic setting and background information for the problem addressed in the paper. Section 3 introduces notation and specifies the assumptions of the model. Section 4 describes the knowledge updating process when new information is obtained from screening. Section 5 defines the screening prioritization problem, describes an exact solution method, and proposes several heuristics. Section 6 evaluates and compares screening heuristics. We provide insights and concluding remarks in Section 7.

2. Setting

A processor of intelligence data has to screen a given set of *intelligence items*, henceforth called simply *items* for brevity. Each item is the content of an intercepted communication, such as e-mail or telephone conversation, between two *participants*. The items induce an *intercepted network* in the following manner. The participants are nodes of the intercepted network, and an edge is

present between two nodes when there is at least one item associated with the corresponding participants. Given an intelligence query–a concrete question posed by an intelligence analyst or a decision maker–each item is either *relevant* to the query or *irrelevant*. While in general there could be multiple levels of relevance, we consider, for simplicity, a binary setting – an item is either relevant or irrelevant for the query. The processor's task is to identify as many relevant items as possible during a limited time period, which is insufficient for screening all the items in the set. The screening times of the different items may not be the same. However, these times are usually unknown a-priori and therefore not considered in determining the screening sequence of items. Thus, the objective is to determine the screening sequence of a given length that maximizes the expected number of identified relevant items. The screening sequence may be adaptive, in the sense that the choice of the ith item to be screened depends on the revealed relevance of the previously screened $i-1$ items.

In addition to the relevance of an item, we also consider the *value* of a participant as a source of relevant information. This value indicates the likelihood or the propensity in which the participant is involved in a relevant communication. Unlike the binary nature of the relevance of an item, we assume that a participant's value can have multiple values ranging from 0 (no value as an information source) to 1 (top value). The value measures of two participants affect the probability that an item between them is relevant. This relation is formally defined in Section 4.

Before beginning the screening, the processor is provided with some partial information, originated by other intelligence sources, regarding the participants identities, occupations and past records as information providers. This information generates a prior probability distribution for the value of each participant. These distributions may range from uniform, when there is no information, to complete certainty regarding the value of a participant, when the processor is certain of the participant's identity and value as information provider for the query. In this latter case, the participant is said to be *identified.* Otherwise the participant is said to be *unidentified* and his value is only known in probability.

The screening process proceeds in rounds. At each round, the processor selects an unscreened item and screens it. The result of the screening is a clear identification of the item as relevant or irrelevant. While false positive and false negative identifications are possible, we assume here no such errors. The generalization to error-prone screenings will be treated in future work. In addition to revealing the relevance of the screened item, screening could also reveal the value of one or both participants associated with the item, even if the item is irrelevant. For example, in a conversation that has nothing to do with the query (i.e., irrelevant item), one of the participants may mention something (name, date, place, etc.) that will immediately reveal the identity and value of one or both of the participants, who thus become identified. We call such an identification, caused by a

crucial piece of information unexpectedly present in a screened item, a *sudden revelation*. As items are screened and knowledge is gained by the processor, the probability distribution of the value of a participant is updated in a Bayesian manner, as described in Section 4.

Assuming independence, the relevance of items associated with a certain pair of participants form a random sample from a Binomial distribution. If the parameter (probability) of that distribution is known to the processor in advance, for each pair of participants, then the optimal screening decision is a greedy one – always select an item that has a maximum probability of being relevant. However, this parameter is initially unknown to the processor and therefore is treated as a random variable whose probability distribution is updated as knowledge is gained during screening.

Finally, we note the difference between items and participants regarding information gathering: the relevance of an item is always revealed after it is screened, while the value of a participant is either revealed through sudden revelation or is updated in a Bayesian manner. As the screening progresses and information is gained, the processor can make better screening decisions based on all information gathered thus far.

3. Notation and Assumptions

Let $G=(V,E)$ denote the graph of the intercepted network. The nodes of the graph, V, represent participants and an edge $(u,v)\in E$ exists if and only if participants u and v generate at least one item. With each edge $(u,v)\in E$, we associate a set containing all the items in which u and v are the two participants. Let $q(e)$ be the subset of items associated with edge $e\in E$.

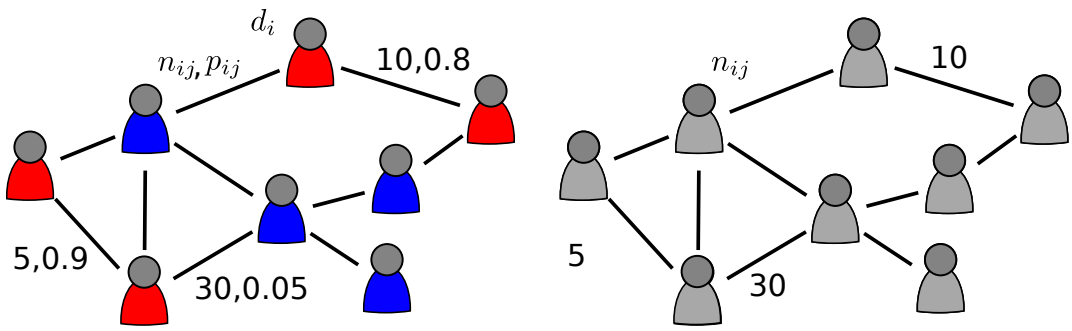
Let $d_u\in[0,1]$ be the value of participant u and let p_e be the probability an item in the subset $q(e)$ is relevant. While it is quite natural to assume that d_u takes a finite number of values (e.g., very valuable, valuable, moderately valuable, etc.) it is clear that p_e can take, in principle, any value in $[0,1]$. However, to simplify the exposition of the model, we assume that p_e also takes a finite (possibly, very large) number of values. As mentioned above, we assume that, given p_e, the items in $q(e)$ form a random sample from a Binomial distribution with probability parameter p_e. The values of the parameters $d_u,\ u\in V$, and $p_e,\ e\in E$, together with the topology of G and the subsets $q(e)$, $e\in E$, represent the ground truth associated with the set of participants and items. As mentioned earlier, if the values of p_e are known to the processor for each $e\in E$ then the optimal screening process is a greedy one – always screen an item from an edge e such that p_e is maximal – and therefore the knowledge regarding the d_u values is redundant. However, the values of p_e and d_u are not known with certainty – the processor only has a perception of this ground truth manifested in probability distributions – and therefore the screening process is not trivial. While the probability of sudden revelation by screening an item in $q((u,v))$ may depend on the true values d_u and d_v, for simplicity we assume that such events occur independently for the two nodes, each with a fixed probability c.

Let P_e be an unobserved random variable whose distribution represents the processor's belief regarding the value of p_e. Similarly, let D_u be a random variable representing the belief regarding the value of d_u. The random variable D_u may only be observed by sudden revelation, otherwise it is also unobserved. In addition to knowing the graph topology and the number of items associated with each edge, the processor has some prior information manifested in the joint distribution of the random vectors $\overline{P} = (P_1, ..., P_{|E|})$ and $\overline{D} = (D_1, ..., D_{|V|})$. We assume that this prior information, which is based on past experience and exogeneous intelligence input, includes the following:

1. Conditional probability distribution of relevance of an item, given the values of the item's participants. Formally, $\Pr[P_{uv} \mid D_u, D_v] \doteq \Pr[P_{uv} = p \mid D_u = d_u, D_v = d_v],\ u, v \in V$.
2. The random variables $\overline{D}$ form a Markov random field. More specifically, let $\overline{D}_A = (D_u, u \in A)$ for some $A \subset V$. The Markov random field property states that any two values D_u and D_v of nodes u and v, respectively, are independent given that $\overline{D}_A = \overline{d}_A$, for a cut-set A separating u from v in the graph G. A cut-set A separates two nodes u and v if all paths from u to v pass through at least one node in A. The operational meaning of this assumption is that dependency between two participants is only attributed to the participant's observed connections in the topology of G.
3. Prior joint probability distribution of $\overline{D}$ is given. The Hammersley-Clifford theorem (Koller and Friedman 2009) allows this distribution to be specified through potential functions on the maximal cliques of G. In particular, this means that for any clique of participants, the processor has a potential function, which can be interpreted as the joint probability distribution of the values of the clique's participants if they were in a graph solely by themselves. Formally, for any maximal clique $C \in G$ the potential function $\Phi_C(\overline{D}_C)$ is given and therefore the distribution of $\overline{D}$ is given.

The model comprises two parts. The first part addresses the knowledge updating process following a screening of an item. The result of such a screening is identifying the item as either relevant or irrelevant, and perhaps observing some sudden revelations. This information is used for updating the joint probability distribution function (pdf) of $[\overline{P}, \overline{D}]$, which we denote as $\Pr[\overline{P}, \overline{D}]$. The second part utilizes the updated joint pdf to determine the next item to be screened. Figure 2 depicts the ground truth associated with the intercepted network and the prior knowledge of the processor regarding that network.

From a modeling perspective, the last assumption above is the strongest. When discussing this application area with intelligence analysts, they are often worried about missing links between people. In other words, they are worried about people who communicate or are closely associated, but somehow the intercepted data simply does not include those connections – perhaps because they use an unknown physical intermediary, or other non-technological means of communication.



(a) Ground truth (b) Initial processor knowledge

Figure 2 **A graphical depiction of a model for intelligence processing: (2(a)) represents ground truth (2(b)) represents initial processor knowledge. The available intelligence items induce a graph where the nodes are participants. With each edge in the graph, we associate the intelligence items between the corresponding participants. For example, the top right edge is associated with 10 conversations. In general, there are n_{ij} conversations on edge (i, j). Some participants have low value, presented here in blue, and some have high value, presented here in red. In general, the value of a participant i is d_i. The probability that a conversation on edge (i, j) is relevant is p_{ij}. For example, the edge on the top right has an 80% chance of producing a relevant conversation upon screening, while the edge on the bottom left has a 90% chance. In general, the processor only knows 1) the graph topology 2) the number of items on each edge and 3) a joint probability distribution $\Pr[\overline{P}, \overline{D}]$ representing beliefs on the values of p_{ij} and d_i.**

The Markov assumption above assumes that people for whom we have no observed connections are conditionally independent, given a separating set of individuals in the social network. This may not be true if there are unobserved connections. This modeling detail can be expanded further. For example, in some sense, people who communicate often – say we have 10000 intercepted communications– should be more dependent than those who communicate less often – say where we have only 1 intercepted communication. Further improvement can be made on the above model by incorporating either of these aspects: 1) Unobserved connections that are missing from the intercepted data and 2) A graceful fade in dependence between individuals who communicate often, and those that do not.

4. Updating

From the three assumption given in Section 3, and the Hammersley-Clifford theorem we obtain that the joint probability distribution $\Pr[\overline{P}, \overline{D}]$ is given by

$$\Pr[\overline{P}, \overline{D}] = \frac{1}{Z} \prod_{C \in \mathcal{C}} \Phi_C[\overline{D}_C] \prod_{(u,v) \in E} \Pr[P_{uv} \mid D_u, D_v],$$

where $\mathcal{C}$ denotes the set of maximal cliques in G, and Z is a normalizing constant. Note that the marginal distribution of $\overline{D}_C$ obtained from $\Pr[\overline{P}, \overline{D}]$ need not be equal $\Phi_C[\overline{D}_C]$. This is the reason

for referring to $\Phi_C[\overline{D}_C]$, $C \in \mathcal{C}$, as clique potential functions rather than pdfs. However, if C is the only maximal clique in the graph, then a normalized $\Phi_C[\overline{D}_C]$ gives the joint probability distribution of $\overline{D}_C$.

Next we show how, in general, the joint probability distribution $\Pr[\overline{P}, \overline{D}]$ is updated as new items are screened. Later on we adjust this general procedure to our specific setting. Let $S_a = 1$ if item a on edge (u, v) is relevant and 0 otherwise, and let $\overline{S} = (S_a, a \in q(u, v), (u, v) \in E)$. The joint distribution of $[\overline{P}, \overline{D}, \overline{S}]$ is

$$\Pr[\overline{P}, \overline{D}, \overline{S}] = \frac{1}{Z} \prod_{C \in \mathcal{C}} \Phi_C[\overline{D}_C] \prod_{(u,v) \in E} \Pr[P_{uv} \mid D_u, D_v] \prod_{a \in q(u,v)} \Pr[S_a \mid P_{uv}], \tag{1}$$

where $\Pr[S_a \mid P_{uv}]$ is simply P_{uv} if $S_a = 1$ and $(1 - P_{uv})$ if $S_a = 0$. When the processor screens an item, say item $\hat{a} \in q(u, v)$, and discovers that it is relevant, he updates his belief distribution to

$$\Pr[\overline{P}, \overline{D}, \overline{S} \mid S_{\hat{a}} = 1].$$

The update for identifying an irrelevant item is similar. In addition to the relevance of screened items, the processor may experience a sudden revelation that reveals the true value of a participant. For simplicity, sudden revelations occur independently for each participant involved in the screened conversation with a fixed probability c, though this can be easily extended to depend on the values of the participants and to participants not involved in the screened conversation. An update for sudden revelations take a similar form, that is, $\Pr[\overline{P}, \overline{D}, \overline{S} \mid D_u = d]$, if participant u is revealed to have a value of d.

Computationally, a naive implementation of the updating process is intractable for large networks. For example, if D_u and P_{uv} are discrete with only two values, a naive method to store the joint distribution $\Pr[\overline{P}, \overline{D}, \overline{S}]$ would require storage of $2^{|V|+|E|+|I|}$ values, where $|I|$ is the number of items. Even with efficient storage of the joint probability distribution, exact computation of the marginal distributions is #P-complete (Dechter 1996) . Even approximate computation of the marginal distributions is, in general, NP-hard (Dagum and Luby 1993). A number of algorithms exist for computing the exact marginal distributions of the unobserved variables, given the values of the observed variables including variable elimination (Zhang and Poole 1996), conditioning (Shachter et al. 1994), and clique trees (Shafer and Shenoy 1990, Shenoy and Shafer 2008). These algorithms, called collectively *inference algorithms*, have been shown to be computationally equivalent, with some trade-offs (Shachter et al. 1994). In general, the running time of inference algorithms is exponential in the tree-width of an appropriate graphical representation of the dependencies in the probability distribution (Koller and Friedman 2009, pp.308-311). Algorithms for computing the marginal distributions approximately, based on sampling estimation, also exist, and

are provably sub-exponential if the conditional probabilities in the model are bounded away from zero (Dagum and Luby 1997). A good overview of algorithms and their associated complexity is provided by Koller and Friedman (2009) .

The efficiency of the inference algorithm used to compute the updated marginal distributions does play a role in the scalability of the intelligence processing model we develop, and it has been shown that inference algorithms work well in large networks – with sizes greater than 4500 nodes (Murphy et al. 1999). However, the details of efficient inference implementations are beyond the scope of this work. Our focus is on developing an efficient intelligence processing scheme, and thus assume that there exists an efficient inference algorithm, relative to the time scale of the processing stage, for computing marginal distributions. We demonstrate computational results for our model in Section 6.

Utilizing the structure and assumptions of our problem, we observe that an inference algorithm only needs to deal with the random variables $\overline{D}$ because, according to Assumption 1 in Section 3, the probability distribution of $\overline{P}$ is derived from $\overline{D}$. Following an observation on edge (u,v), S_{uv}, the sequence of operations is much trimmer than the general mathematics described above. Specifically, the process comprises four stages:

1. Update $\Pr[P_{uv}]$:

$$\Pr^U[P_{uv}=p]=\Pr[P_{uv}=p\mid S_{uv}=s]=\begin{cases}\frac{p\Pr[P_{uv}=p]}{E[P_{uv}]} & \text{if } s=1\\ \frac{(1-p)\Pr[P_{uv}=p]}{1-E[P_{uv}]} & \text{if } s=0\end{cases}$$

   where $\Pr[\cdot]$ and $\Pr^U[\cdot]$ are prior and posterior (updated) probability distributions, respectively.
2. Update $\Pr[D_u,D_v]$:

$$\begin{aligned}\Pr^U[D_u=d_u,D_v=d_v]&=\\ &=\sum_p \Pr[D_u=d_u,D_v=d_v\mid P_{uv}=p]\Pr^U[P_{uv}=p]\\ &=\sum_p \frac{\Pr[P_{uv}=p\mid D_u=d_u,D_v=d_v]\Pr[D_u=d_u,D_v=d_v]}{\Pr[P_{uv}=p]}\Pr^U[P_{uv}=p].\end{aligned}$$

3. Use an inference algorithm along with the updated distribution $\Pr^U[D_u,D_v]$ to derive updated marginal distributions of the other pairs of nodes (k,l), $\Pr^U[D_k,D_\ell]$.
4. The updated marginals $\Pr^U[D_k,D_\ell]$ give updated distributions $\Pr^U[P_{k\ell}]$ for $(k,\ell)\neq(u,v)$:

$$\Pr^U[P_{k\ell}=p]=\sum_{d_k,d_\ell}\Pr[P_{k\ell}=p\mid D_k=d_k,D_\ell=d_\ell]\Pr^U[D_k=d_k,D_\ell=d_\ell] \tag{2}$$

The above steps show that an inference algorithm is only required for updating the distribution of the variables $\overline{D}$, allowing for faster computation.

5. Screening

The decision problem faced by the processor is how to sequence screening in a given number rounds, T, so as to maximize the expected number of identified relevant items. In this section, we formalize the processor's screening prioritization problem and propose several heuristics for generating effective screening sequences.

For simplicity of exposition, we assume that the value d_u of each participant — i.e. node in the social network — is either 0 or 1. It is straight-forward to extend this binary setting to multiple discrete quantities. The formal statement of the *screening prioritization problem* is as follows. Let the processor's initial belief distribution $\pi_0 = \Pr[\overline{D}]$ be given. The distribution π_0 along with the conditional distribution assumptions is sufficient to obtain the joint distribution of all random variables as in equation (1). Let δ_i specify the item to be screened at round i. Formally, δ_i is a function outputting an item to be screend with inputs π_0, the initial belief distribution, and h_{i-1}, the history of items screened thus far. This history describes the relevance values and sudden revelations of screened items in rounds previous to round i. The screening optimization problem is:

$$\max_{\delta_i} E[\sum_{i=1}^{T} S_{\delta_i(\pi_0, h_{i-1})}], \tag{3}$$

where $S_{\delta_i(\pi_0,h_{i-1})}$ is the relevance value of the item screened on round i.

The screening prioritization problem, (3), is a finite state, finite action space, finite horizon partially observable Markov decision process (POMDP) (Monahan 1982). While problem (3) is stated generally, in practice the processor only needs to select an edge rather than an item, because the relevance of all items in $q(e), e \in E$, is assumed to be equally likely. Following the POMDP definition of Monohan (Monahan 1982), the POMDP to be solved by the processor is defined by:

1. A core process with states $(\overline{\ell}, \overline{p}, \overline{d})$, where $\overline{\ell} = (\ell_e, e \in E)$ is a vector of non-negative integers representing the number of unscreened items remaining on each edge, $\overline{p}$ represents the probabilities that an item on each edge is relevant, and $\overline{d}$ represents the values of the participants. The core process's state is not entirely observed by the processor, he only knows $\overline{\ell}$. The values of $\overline{p}$ and $\overline{d}$ are only known in probability.
2. A probability distribution, π_0, known to the processor, specifying an initial distribution on core states.
3. An action space of the processor, E. In each round he selects an edge from which to screen an item.
4. Transition probabilities from one the core state to another, given a selected action $e \in E$. For this POMDP, the transition probabilities are simple. Most coordinates of the core state $(\overline{\ell}, \overline{p}, \overline{d})$ do not change, with the exception of ℓ_e, which decreases by one deterministically, for the selected edge e.

5. Observations on the core process state, defined by the relevance of the screened conversation, and any sudden revelations that occur.
6. A likelihood of seeing observation y, given the core process is in state x. This is given by p_e, the probability of observing a relevant conversation on edge e, and c, the probability of a sudden revelation for either of the participants.
7. Finally, a reward function, which is 1 if a relevant conversation is observed and 0 otherwise.

It is well known that a POMDP can be converted into an equivalent Markov decision process (MDP) (Monahan 1982), where the state space of the MDP is the space of probability distributions on the core states of the POMDP. Specifically, the state in round i of the equivalent MDP is a probability distribution on $(\overline{\ell}, \overline{p}, \overline{d})$ – given by the certain knowledge on the number of unscreened conversations, $\overline{\ell}$, and a belief distribution on $\overline{p}, \overline{d}$ given by π_0 updated with knowledge in h_{i-1}.

With this equivalence, dynamic programming gives an exact, though inefficient, algorithm for maximizing the expected reward – the expected number of identified relevant items over T rounds. Though exact, the prohibitive running time of dynamic programming, $O(|E|^T \cdot \text{ Infer})$ where Infer is the inference algorithm run time, leads us to seek heuristic optimization methods for the processor's screening problem.

In a nutshell, the basic tension in the screening prioritization problem is one between exploitation and exploration. *Exploitation* screens edges known to have a relatively high probability of yielding relevant conversations — i.e., high $E[P_{ij}]$. *Exploration* seeks information about relatively unknown areas of the social network — i.e. provides more precise estimates of $E[P_{ij}]$ for the graph's edges. The exploration vs. exploitation tension is similar to the tension encountered in multiarmed bandit problems, which date back to the 1950s and continue to engender scientific study today (Robbins 1952, Jones and Gittins 1972, Auer et al. 2002b,a, Dar et al. 2002, Mannor and Tsitsiklis 2004), with two key exceptions. First, in the screening prioritization problem, edges, which correspond to bandits, are correlated. Information about one bandit provides information on others. Second, bandits have a limited number of pulls. In particular, we can only select edge e for screening as many times as the number of items in $q(e)$. Nevertheless, the multiarmed bandit literature provides a rich set of heuristics that may be naturally applied to the screening prioritization problem.

The screening prioritization problem is related to two research areas: contextual bandit models, and sequential exploration problems. Contextual bandit models are often used for suggesting articles and other content to internet users. In a contextual bandit model, the reward of an action – the match to a user's interest – depends on a context presented to the algorithm. The context can be thought of as a description of the user's interests, which allows for computation of some kind of a matching index. However, contextual bandit approaches neither contain any underlying social network structure, nor they address the informational value of pairs of nodes. Furthermore, some

of the most advanced algorithms for contextual bandits depend on a linear relationship between the context and the expected payoff of the actions (Wang et al. 2005, Li et al. 2010, May et al. 2012). Our approach is starkly different, in that it concerns an explicit social network structure that results in non-linear relations between the arms (edges). Sequential exploration problems have been applied in the search for oil and gas reserves (Martinelli et al. 2012, Brown and Smith 2012). These problems are similar to ours in that there is a complex Bayesian graphical model in the background that captures the knowledge state of the decision maker, as more information is collected through executing actions. Our problem differs from typical sequential exploration in its underlying network structure and in the specific way knowledge is accumulated and updated. In our model, the arms of the bandit (edges in the social network) are correlated in a unique way that depends on the topology of the network and the values of the nodes, which are updated dynamically. Because of this unique setting, even the most basic heuristic discussed below, Pure Exploitation, requires, in our case, a complex inference process. Other studies, that assume a specific correlatation structure amongst the bandits, also exist (Rusmevichientong and Tsitsiklis 2010, Frazier et al. 2009).

5.1. Heuristic Algorithms for the Screening Optimization Problem

We first explore two simple and efficient heuristics: *Pure Exploitation (PE)*, and *Softmax* (Daw et al. 2006). Then we investigate two additional, more complex, heuristics: *Value-Difference-Based-Exploration (VDBE)* (Tokic 2010) and a *Finite Horizon MDP (FHM)* policy. We describe these four heuristics in turn.

Pure Exploitation (PE) is a greedy algorithm that always chooses an item from the edge with the highest expected probability of being relevant, i.e., an edge (u, v) with the highest value of $E[P_{uv}]$. This algorithm ignores exploration altogether, and always chooses an item according to the exploitation criterion. This naive approach can serve as a baseline for comparison with other algorithms as it is exactly optimal when $Var[P_{uv}] = 0$ for all $(i, j) \in E$. Though PE's selection criterion for an edge is simple, it still depends on updating the knowledge state of the processor after each round.

Softmax implements a mixture of exploration and exploitation (Thrun 1992). The algorithm assigns a weight to each edge, and selects an edge randomly proportional to those weights. The weight on edge (u, v) is $w_{uv} = e^{\frac{E[P_{uv}]}{K}}$, where K is a positive constant often called temperature (Daw et al. 2006). For small values of K, the weight of edges with high $E[P_{uv}]$ is large and they will almost always be chosen, favoring exploitation over exploration. For large values of K, all variables have approximately the same weight and random exploration is dominant.

Value-Difference-Based-Exploration (VDBE) (Tokic 2010) is a modification of a classical ϵ-greedy algorithm that mixes exploitation and exploration probabilistically. The value of ϵ specifies

the probability that exploration is chosen and may be a constant parameter or a function of the remaining available screening time (Tokic 2010). VDBE explores when there is a low certainty regarding the expected reward of alternative actions, and exploits otherwise. The expected reward of screening edge (u, v) is $E[P_{uv}]$. The algorithm updates the likelihood of exploration according to the formula:

$$\epsilon^{k+1} = \delta \frac{1 - e^{\frac{-U}{\sigma}}}{1 + e^{\frac{-U}{\sigma}}} + (1 - \delta)\epsilon^k,$$

where U is the maximum difference in expectations between the $(k-1)$st screening and the kth screening, $U = max_{uv}|E^k[P_{uv}] - E^{k-1}[P_{uv}]|$ with expectations taken with respect to the updated probability distributions in the respective rounds. The parameters δ and σ vary the long-term behavior of the algorithm. The value of ϵ^0 is set to 1.

Finite Horizon MDP (FHM) can be thought of as a type of a Knowledge-Gradient policy (Frazier et al. 2009). The prohibitive run time of the exact dynamic programming algorithm, $O(|E|^T \cdot \text{Infer})$, comes from the fact that the only state values known are those at the final time horizon, T. Because of this, the exact algorithm must look T rounds into the future to compute the optimal actions. The efficient but inexact FHM algorithm works by employing an estimate of the state value at a certain depth of the dynamic programming algorithm. Consider a state of the dynamic program defined by (π_0, h_i) , with $T - i$ rounds of screening remaining. One reasonable estimate of the state's value comes from assuming that belief distribution updates are no longer performed, giving a state value estimate of $\sum_{a \in A} E[P_{e_a}]$, where A is the set of $T - i$ most likely relevant items under $\Pr[\overline{P}, \overline{D} \mid \pi_0, h_i]$ and e_a is the edge of item a. Such an estimate function, combined with a finite horizon dynamic program gives a family of optimization algorithms, based on the lookahead depth, that trade off optimality and efficiency. Another way to look at these algorithms is to consider them as a type of rolling, fixed-exploration period algorithm. The algorithm has a certain period which it can explore–the lookahead depth–and after that it must exploit based on the knowledge it gained. This bears resemblance to the budgeted learning problems that appear in the literature (Guha and Munagala 2007). For simplicity, we fix the lookahead depth to one.

In the next section we study these heuristics and compare their performance. We select these heuristics in particular for the following reasons. Pure Exploitation is the natural, most simple algorithm to try. Softmax and ϵ-Greedy are two commonly used ad-hock methods for multiarmed bandit problems that mix exploration and exploitation, unlike Pure Exploitation (Vermorel and Mohri 2005). Some recent results show that VDBE has more robust performance on multiarmed bandit problems than Softmax and classical ϵ-Greedy (Tokic 2010). Finally, the FHM heuristic is included because it provides a theoretically optimal solution, as long as the number of items to be screened is smaller than the heuristic's horizon.

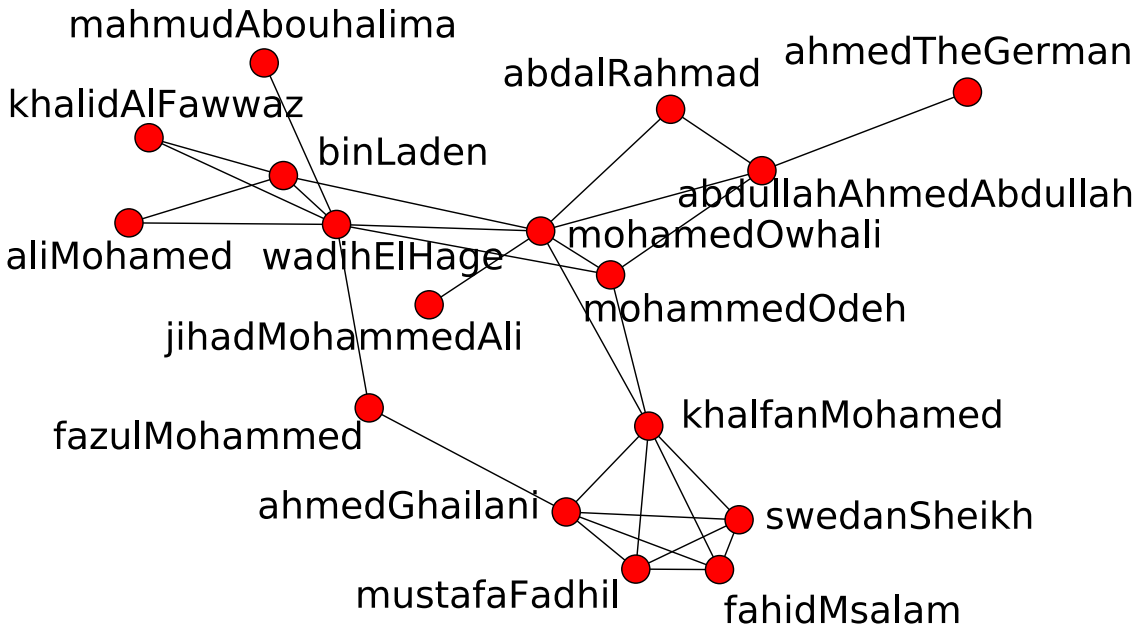


Figure 3 **The nodes in the graph are 17 terrorists behind the 1998 US embassy bombing in Tanzania. The links in the graph represent known associations. The data for this network is from the Carnegie Mellon Computational Analysis of Social and Organizational Systems laboratory (Computational Analysis of Social and Organizational Systems (CASOS) 2012) .**

6. Analysis

Our computational example, in which we test the aforementioned screening heuristics, is based on a real-world social network of 17 terrorists who were behind the 1998 United States embassy bombing in Tanzania (Computational Analysis of Social and Organizational Systems (CASOS) 2012). The terrorist social network is depicted in Figure 3. We augment the graph of the social network with 17 benign nodes, along with randomly generated adjacent edges, reflecting possible communications among the 17 non-terrorist nodes and between them and the terrorists.

The ground truth is such that the value of a node d_u is fixed at 1 for each one of the 17 terrorists nodes and at 0 for all 17 benign nodes. The probabilities of relevant items, p_{uv}, are selected randomly from a Beta distribution on P_{uv}, depending on the values of d_u and d_v . We choose to use the Beta distribution because its support is bounded by 0 and 1, as needed for P_{uv}, and more importantly, it is the conjugate prior for the Bernoulli observations. This guarantees that Bayesian updates keep P_{uv} as a Beta distributed random variable, which makes the analysis tractable. The complete details of the computational implementation are given by Nevo (Nevo 2011). Figure 4 depicts the parameterized network.

Some of the heuristics contain parameters, such as the scaling parameter K in Softmax, that require tuning. We tune each algorithm, by testing it several times with different parameter combinations, and select the parameter combination that yields the best performance. However, in a

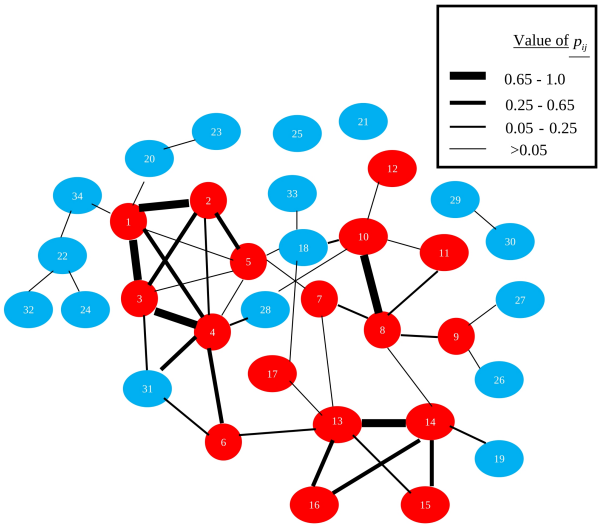


Figure 4 **The parameterized network used in the computational example. The network includes benign nodes, pictured in blue, and terrorists, pictured in red. The network also includes a specific likelihood of relevant items on each edge. The edges are grouped in ranges, based on their likelihood, pictured here by a different edge thickness for each range. The number of items on each edge is varied according to a Poisson distribution with a specified mean.**

practical application, a processor does not have the benefit of trying the algorithm many times on the screening prioritization problem instance of interest to select the best parameters. In a realistic situation, either careful apriori reasoning is required for algorithm parameter tuning, or algorithms with small numbers of parameters, such as PE, Softmax, or FHM, are to be used. PE, for example, has no parameters and thus does not even require apriori reasoning for effective use. Softmax has one parameter, called *temperature*, that represents a relatively unintuitive way of specifying what fraction of iterations the algorithm uses for exploration as opposed to exploitation. Tuning Softmax's *temperature, K,* requires estimating the ratio of $\frac{e^{\frac{\max_{ij} p_{ij}}{K}}}{e^{Avg_{ij}[p_{ij}]_K}}$, an unintuitive and possibly instance-dependent task. In contrast, FHM, has a very intuitive single parameter – the number of lookahead steps. Further, experimentation on several instances shows that a lookahead of one step is about as good as larger lookaheads. Thus, the single parameter could potentially be just fixed at one-step lookahead for all instances. Based on our experimentation, in terms of parameterization, the clear winners are PE and FHM for their ease of interpretation and parameterization. For further details about parameter tuning, see Nevo (2011).

Figure 5 illustrates the behavior of the heuristics. Each figure presents a single run of $T = 300$ screenings and a mean of 100 items per edge in the network shown in Figure 4. Recall that p_e denotes the true (simulated) probability that an item on edge e is relevant. We define the *distance* of an edge selected for screening, w, to be $p_{e^*} - p_w$, where $e^* = \arg\max_e \{p_e,\ q(e) \text{ is not empty}\}$. In other

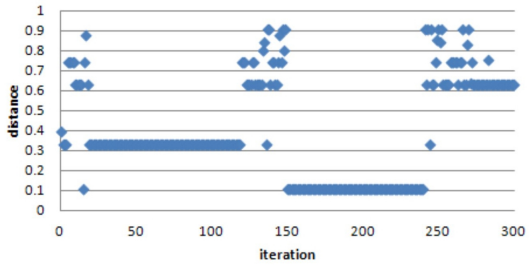
words, e^* is an edge with unscreened items that has the highest probability of yielding a relevant item. Edges with distance zero are optimal when it comes to yielding relevant conversations. Edges with a higher distance values are farther away from the optimal selection. The vertical axis of the figure represents the distance of the edges selected by the algorithm; in other words, the plots indicate how close each screening selection is to an optimal one. The horizontal axis represents the 300 screenings in the single run.

Pure Exploitation always selects the edge with highest $E[P_{ij}]$. An update in the distribution of $\overline{P}$ may result in selecting a different edge in the following screening round, as seen in the first two screenings in the graph in 5(a) where the distances change. From approximately the $20th$ round and on, the same edge is repeatedly selected until all items on that edge are exhausted, which explains the flat line at that range. Observations from this edge do not sufficiently change the belief distribution to change Pure Exploitation's selection.

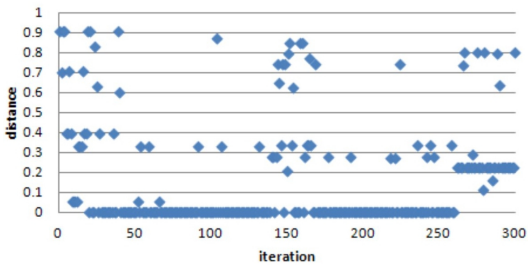
The behavior of Softmax, shown in Figure 5(b), is different than that of Pure Exploitation. Softmax selects edges probabilistically, so that there is always some chance for selecting an edge that does not have the maximum value of $E[P_{ij}]$; choosing exploration over exploitation. Softmax's exploration identifies optimal or near-optimal edges for screening, as indicated by the long sequence of distance zero selections. Even though the algorithm finds a near-optimal edge, the algorithm always spends some time exploring, as indicated by the positive distance edges selected throughout the run. In these single runs, Softmax's exploration allows it to generally select better edges – edges with a better relevance probability p_e, than Pure Exploitation.

The behaviours of the two remaining heuristics are less intuitively simple, but consisent with the literature. For exmple, VDBE enters a series of exploration steps whenever it encounters an edge that performs worse than the algorithm expected – because the value difference as defined by the algorithm is great. FHM tends to explore early, because its finite horizon makes the algorithm behave as though it is not possible to explore later in the screening process.

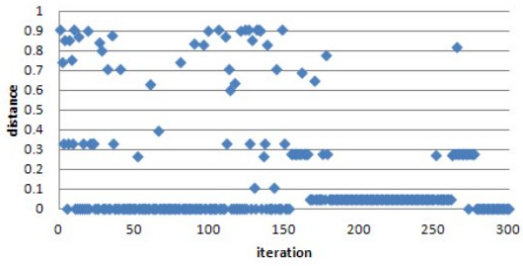
Figure 6 compares the performance of the four heuristics to the performance of a greedy algorithm in the ideal situation when all the p_e values are known. As mentioned earlier, the latter this is the best possible screening situation one could expect and therefore it represents an upper bound for the performance of any screening procedure. Each bar represents 150 simulation runs, each for $T = 300$ screenings. Blue, red, and green bars represent the means of 50, 100, and 350 items per edge, respectively. The figure shows several surprising results. First, even though a complex algorithm like FHM has the best performance among the suggested heuristics, simple algorithms like Pure Exploitation and Softmax perform surprisingly well. Second, when there is a small number of items per edge – a situation represented in the blue bars – all algorithms perform essentially



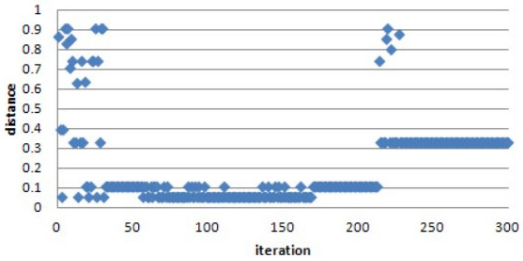
(a) Pure Exploitation Behavior



(b) Softmax Behavior



(c) VDBE Behavior



(d) FHM Behavior

Figure 5 **Example behaviors of the heuristic algorithms. Figure 5(a) shows the behavior of Pure Exploitation. The vertical axis, labeled distance, is the difference $p_{e^*} - p_w$, where e^* is the edge with the greatest likelihood of yielding a relevant conversation and w is the edge selected by the algorithm. In other words, if the distance is 0, the algorithm is selecting the best possible edge to screen, and larger values indicate that the algorithm is selecting poorer edges. The plot represents a single run with 300 screenings, as indicated by the horizontal axis. The remaining figures represent example behaviour of the other heuristics.**

the same. Third, the best performing algorithms – Softmax and FHM – show surprisingly little variance in performance.

The run times of the algorithms depend on two parameters: The run time to select an edge, and the run time to perform inference. The only algorithm that has significant run time in selecting an edge is FHM because during its lookahead phase it must perform several inferences. Ultimately, the bottleneck in all the run times is the inference on the graphical model. The inference run times themselves are highly dependent on the structure of the graph. We have constructed instances where the inference from a single screening takes as much as 2 minutes on a modern Intel Xeon processor! On the other hand, if the graph is almost a tree, the inference can take less than half a second for large graphs, with a hundred nodes. This difference in inference run time is due to the fact that exact inference on a graphical model is exponential in the tree-width of the graph. It also suggests a promising line of research in using approximate inference to speed up these times on graphs that have large tree-width.

This computational analysis, and further analysis detailed in (Nevo 2011) , leads to several key insights:

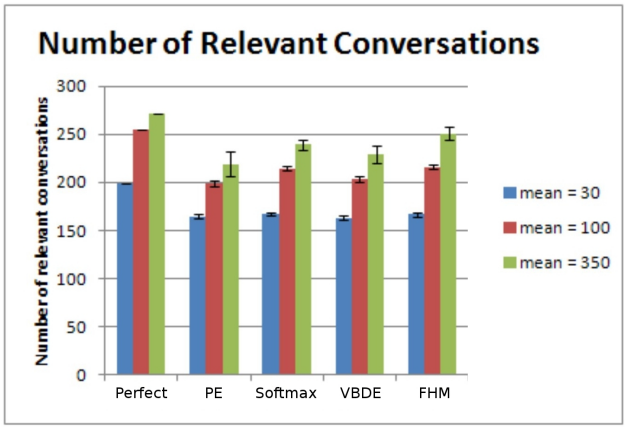


Figure 6 **Algorithm performance. Perfect denotes a greedy selection algorithm that apriori knows the p_e values. Perfect provides an upper bound on the performance of any other algorithm. The other algorithms presented are: PE, Softmax, VBDE, and FHM. Each color bar comes from a 150 algorithm simulation runs, each for 300 screenings. Blue, red, and green bars represent the means of 50, 100, and 350 items per edge, respectively. The error bars on top of each bar represent standard deviation in algorithm performance.**

1. *Simple algorithms perform surprisingly well.* Even a simple algorithm like Pure Exploitation performs reasonably well. We remark that even though Pure Exploitation is in itself simple algorithm, its performance depends on knowledge updates, which themselves are a complex computational process. Softmax, which is also a rather simple algorithm, performs nearly as well as more complex algorithms. We speculate that the dependencies among the alternative actions, the edges available for screening, are the main reason for this good performance.
2. *Exploration is less valuable in an exploration-exploitation setting where pulling one bandit gives information on several others as well.* One could think of the screening prioritization problem as a multi-armed bandit problem with correlated bandits – each edge corresponds to a bandit – where pulling one bandit yields information on many others. In this setting, exploration is less valuable than in an uncorrelated bandit setting, as demonstrated by the similar performance of Pure Exploitation and Softmax.
3. *Exploration is less valuable in an exploration-exploitation setting with a limited number of visits to each bandit.* In the screening optimization problem, a limit on the visits to each bandit is presented by the number of items on each edge. If the visits to a bandit are limited to a small number, it is not worth finding a good bandit because once we find it we have no opportunity to exploit indefinitely. This is demonstrated by the similar performance of all algorithms when the mean number of items per edge is small.

Finally, though computationally expensive, the best algorithm is the FHM policy. This complex algorithm shows consistently good performance, even under varied simulation parameters. However, the simplicity and good performance of Softmax may make it preferable for practical use.

7. Summary and Conclusions

With ever growing sensing and intercepting capabilities that produce a glut of intelligence data, the challenge faced by the intelligence community is to efficiently sort these data, and single out the relevant, and only relevant, data items for a given query. In particular, intelligence data obtained from intercepting communications in a social network, such as terrorists groups or organized crime, may be time-sensitive or even time-critical, to the point that some intelligence items may quickly become obsolete if not processed in a timely manner. Therefore, it is of utmost importance that the intercepted data is processed effectively during the alloted time window. In this paper we propose a method for efficiently screening the intelligence items based on (1) exogenous partial prior information concerning the social network, (2) observed relevance of screened items (communications), and (3) continuously updated information about the values of participants (nodes in the network) as intelligence providers based on the observations of screened items. Using techniques from graphical models, social networks and Bayesian learning we developed a new method for optimally sequencing the screening of items. While the exact optimization scheme is, for all practical purposes, an intractable MDP algorithm, we proposed several heuristics that have been suggested in the literature and examine their effectiveness by extensive simulations. The benchmark is an ideal situation where the relevance probability of each edge in the social network is known to the processor, in which case a simple greedy algorithm is provably optimal. Using these simulations, we show that the expected number of relevant items obtained by simple heuristics are within 80% to 93% of a best-case scenario where the processor has full knowledge about the likelihood of relevant items.

Extensions of this work should address imperfect screening, that is, false negative and false positive screening errors, variable screening time, and various levels of relevance. Variable screening times would significantly alter the model, especially if information on the expected screening time of an item is available prior to screening the item. Intuitively, variable screening times would change the problem from a selection problem, to a stochastic job-shop scheduling problem. On the other hand, if no prior information on the variable screening times is known, the selection problem we present is, in a sense, the best an intelligence analyst can do because all items look the same at the time he selects an item to screen.

Discounting of information in the model can come in two forms: a time discount factor that values information gained earlier in the screening process more than information gained later; and

an information discount, that values the first few relevant intelligence items from a queue (an edge in the network) more than subsequent relevant items from the same queue. Time discounts are less relevant in this application because typically the information has to be screened in a few hours or a day, as opposed to months or years. On the other hand, information discounts are quite relevant because subsequent relevant items from the same queue may contain duplicate information – two people tend to talk about the same topics. An information discount can be included in the model by adding the number of relevant items screened from an edge as part of the POMDP state. Then, subsequent items from the same edge can be valued with exponentially decreasing rewards. Intuitively, such a change would not allow us to exploit edges with high probabilities of relevance and many items because the expected marginal reward from such items would decrease to zero quite quickly.

It is possible to adapt the model we present to multiple simultaneous processors working on the same intelligence collection task. As the multiple processors screen items, they update a common knowledge base of the network. In that sense, the model is similar to the one we present, except the processing is performed at a faster rate. The main difference is that some processing tasks may not yet be complete when selecting a new processing task. For example, if there are five processors, and processors one through four are still screening their items at the time we have to select an item for processor five. Thus, the knowledge updating process may be lagged a bit and the algorithms we present will need to be slightly modified to ignore items currently being processed by others.

Finally, the setting described in this paper is static in the sense that during screening no new items arrive. In a realistic high-intensity environment this may not be the case and intelligence items may arrive continuously over time. This would dynamically change the network structure, and may require a different updating and selecting processes.

References

Auer, P., N. Cesa-Bianchi, P. Fischer. 2002a. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* **47** 235–256.

Auer, P., N. Cesa-Bianchi, Y. Freund, R. E. Schapire. 2002b. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* **32**(1) 48–77.

Brown, D., J. E. Smith. 2012. Optimal sequential exploration: Bandits, clairvoyants, and wildcats. https://faculty.fuqua.duke.edu/ dbbrown/bio/papers/brown_smith_bandits_12.pdf. Accessed on 2013-3-13. Accepted for publication in *Operations Research*.

Computational Analysis of Social and Organizational Systems (CASOS). 2012. Tanzania Embassy CT. http://www.casos.cs.cmu.edu/computational_tools/datasets/internal/embassy/index11.php. Accessed on 2012-10-12.

Dagum, P., M. Luby. 1993. Approximating probabilistic inference in bayesian belief networks is np-hard. *Artificial Intelligence* **60**(1) 141–153.

Dagum, P., M. Luby. 1997. An optimal approximation algorithm for bayesian inference. *Artificial Intelligence* **93**(1-2) 1–27.

Dar, E., S. Mannor, Y. Mansour. 2002. PAC bounds for multi-armed bandit and Markov decision processes. *Proceedings of the 15th Annual Conference on Computational Learning Theory* 255–270.

Daw, N. D., J. P. ODoherty, B. Seymour, P. Dayan, R. J. Dolan. 2006. Cortical substrates for exploratory decisions in humans. *Nature* **441** 876–879.

Dechter, R. 1996. Bucket elimination: a unifying framework for probabilistic inference. *Proceedings of the 12th international conference on uncertainty in artificial intelligence*. San Francisco, CA, USA, 211–219.

Deitchman, S. J. 1962. A Lanchester model of guerrilla warfare. *Operations Research* **10**(6) 818–827.

Frazier, P., W. Powell, S. Dayanik. 2009. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing* **21** 591–613.

Godfrey, G. A., J. Cunningham, T. Tran. 2007. A Bayesian, nonlinear particle filtering approach for tracking the state of terrorist operations. *Intelligence and Security Informatics, 2007 IEEE*. IEEE, 350–355.

Gorman, S. 2002. Probe on christmas plot lists failures. *The Wall Street Journal* .

Guha, S., K. Munagala. 2007. Approximation algorithms for budgeted learning problems. *Proceedings of the thirty-ninth annual ACM symposium on theory of computing*. ACM, 104–113.

Harney, R., G. Brown, M. Carlyle, E. Skroch, K. Wood. 2006. Anatomy of a project to produce a first nuclear weapon. *Science & Global Security* **14** 163–182.

Jones, David Morian, John C Gittins. 1972. *A dynamic allocation index for the sequential design of experiments*. University of Cambridge, Department of Engineering.

Kaplan, E. H. 2010. Terror queues. *Operations Research* **58** 773–784.

Kaplan, E. H. 2012. OR forum: Intelligence Operations Research: The 2010 Philip McCord Morse lecture. *Operations Research* doi: 10.1287/opre.1120.1059.

Koller, D., N. Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series)*. 1st ed. The MIT Press.

Li, L., W. Chu, J. Langford, R. E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th international conference on world wide web*. WWW '10, ACM, 661–670.

Mannor, S., J. Tsitsiklis. 2004. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research* **5** 623–648.

Martinelli, G., J. Eidsvik, R. Hauge, J. Radack. 2012. Dynamic decision making for graphical models applied to oil exploration.
http://arxiv.org/abs/1201.4239. Accessed on 2013-3-13.

May, B. C., N. Korda, A. Lee, D. S. Leslie. 2012. Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research* **98888** 2069–2106.

Monahan, G. E. 1982. A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science* **28**(1) 1–16.

Murphy, K. P., Y. Weiss, M. I. Jordan. 1999. Loopy belief propagation for approximate inference: an empirical study. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. UAI'99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 467–475.

Nevo, Y. 2011. Information selection in intelligence processing. Master's thesis, Naval Postgraduate School, Monterey, CA.

Radack, J. 2011. NSA whistleblowers on 60 Minutes: 9/11 could have been prevented. www.whistleblower.org/blog/31/1128. Accessed on 2012-10-25.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* **58**(5) 527–535.

Rusmevichientong, P., J. N. Tsitsiklis. 2010. Linearly parameterized bandits. *Mathematics of Operations Research* **35**(2) 395–411.

Schaffer, M. B. 1968. Lanchester models of guerrilla engagements. *Operations Research* **16**(3) 457–488.

Shachter, R. D., S. K. Andersen, P. Szolovits. 1994. Global conditioning for probabilistic inference in belief networks. *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*. San Francisco, CA, USA, 514–522.

Shafer, G. R., P. P. Shenoy. 1990. Probability propagation. *Annals of Mathematics and Artificial Intelligence* **2**(1) 327–351.

Shenoy, P. P., G. Shafer. 2008. Axioms for probability and belief-function propagation classic works of the dempster-shafer theory of belief functions. R. R. Yager, L. Liu, eds., *Classic Works of the Dempster-Shafer Theory of Belief Functions*, *Studies in Fuzziness and Soft Computing*, vol. 219. Springer Berlin / Heidelberg, 499–528.

Steele, J. M. 1989. Models for managing secrets. *Management Science* **35**(2) 240–248.

Thrun, S. B. 1992. *Handbook of intelligent control: Neural, fuzzy, and adaptive approaches*, chap. The role of exploration in learning control. Van Nostrand Reinhold, Florence, KY, 527–559.

Tokic, M. 2010. Adaptive epsilon-greedy exploration in reinforcement learning based on value difference. *Proceedings of the 33rd Annual German Conference on Artificial Intelligence*. 203–210.

Vermorel, J., M. Mohri. 2005. Multi-armed bandit algorithms and empirical evaluation machine learning: ECML 2005. J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, L. Torgo, eds., *Machine Learning: ECML 2005*, *Lecture Notes in Computer Science*, vol. 3720. Springer, 437–448.

Wang, C., S. R. Kulkarni, H. V. Poor. 2005. Bandit problems with side observations. *IEEE Transactions on Automatic Control* **50**(3) 338–355.

Zhang, N. L., D. Poole. 1996. Exploiting causal independence in bayesian network inference. *Journal of Artificial Intelligence Research* **5**(1) 301–328.

Zlotnik, J. 1967. A theorem for prediction. *Studies in Intelligence* 1–2.