Social Network Analysis with Content and Graphs

William M. Campbell, Charlie K. Dagli, and Clifford J. Weinstein

Social network analysis has undergone a renaissance with the ubiquity and quantity of content from social media, web pages, and sensors. This content is a rich data source for constructing and analyzing social networks, but its enormity and unstructured nature also present multiple challenges. Work at Lincoln Laboratory is addressing the problems in constructing networks from unstructured data, analyzing the community structure of a network, and inferring information from networks. Graph analytics have proven to be valuable tools in solving these challenges. Through the use of these tools, Laboratory researchers have achieved promising results on real-world data. A sampling of these results are presented in this article.

» **As a consequence of changing economic** and social realities, the increased availability of large-scale, real-world sociographic data has ushered in a new era of research and development in social network analysis. The quantity of content-based data created every day by traditional and social media, sensors, and mobile devices provides great opportunities and unique challenges for the automatic analysis, prediction, and summarization in the era of what has been dubbed "Big Data." Lincoln Laboratory has been investigating approaches for computational social network analysis that focus on three areas: constructing social networks, analyzing the structure and dynamics of a community, and developing inferences from social networks.

Network construction from general, real-world data presents several unexpected challenges owing to the data domains themselves, e.g., information extraction and preprocessing, and to the data structures used for knowledge representation and storage. The Laboratory has developed methods for constructing networks from varied, unstructured sources such as text, social media, and reality mining of datasets.

A fundamental tool in social network analysis that underpins many higher-level analytics is community detection. Various approaches have been employed to detect community structure from network data, including a technique to explore the dynamics of these communities once they are detected. Insight gained from basic analytical techniques, such as community analysis, can be used for higher-level inference and knowledge-discovery tasks. Work in attribute prediction on social networks takes advantage of recent advances in statistical relational learning.

Graph Construction

Social networks are embedded in many sources of data and at many different scales. Social networks can arise from information in sources such as text, databases, sensor networks, communication systems, and social media. Finding and representing a social network from a data source can be a difficult problem. This challenge is due to many factors, including the ambiguity of human language, multiple aliases for the same user, incompatible representations of information, and the ambiguity of relationships between individuals.

Data Sources and Information Extraction

Two primary open sources of social network information are newswire and social media. Various research efforts examine other sources of social network data—smart phones [1–3], proximity sensors [4], simulated data [5, 6], surveys, communication networks, private company data [7], covert or dark networks, social science research, and databases.

Text information from newswire provides information about entities (people and organizations) and their corresponding relations and involvement in events [5]. This information is encoded in text in multiple languages and numerous formats. Extracting entities and their relations from newswire stories is a difficult task.

Sensors can also serve as sources of data for social networks [2]. Smartphones and proximity devices [4] can provide information about dynamic interactions in social networks and can aid in the corresponding analysis of those networks. Predicting behavior, personality, identity, pattern of life, and the outcome of negotiations are a few of the proposed applications that may exploit data from sensor systems.

Communications and social media have been analyzed for social network structure. For example, the release of the e-mail related to Enron's bankruptcy, and subsequent prosecution for fraudulent accounting practices [8], has provided a limited window into company dynamics and e-mail flow. In the case of social media, an analysis of followers in Twitter shows networks of users who are related by current news topics rather than by personal interactions [9]. Other social media companies such as Facebook may also provide network data sources, although privacy is a major concern.

Databases can provide networks in structured form. Examples of database types include transactions between individuals (e.g., bank accounts, Paypal), collaboration or references (e.g., patent, research, movie databases), and human-annotated and -entered data. The research described in this article used a database collected by the Institute for the Study of Violent Groups (ISVG) [10]. This database contains people, organizations, and events annotated and categorized into a standard Structured Query Language (SQL) relational database structure [5].In addition, a database of sensor data from the reality mining corpus [1] is used for dynamic social network analysis.

INFORMATION EXTRACTION FROM TEXT

Information extraction (IE) is a standard term in human language technology that describes technology that automatically extracts structured information from text [11]. A popular subarea in IE is named-entity recognition (NER). NER extracts people, places, and organizations that are mentioned in text by proper name (as opposed to being referenced by pronominal terms, e.g., "you," or nominal forms, e.g., "the man").

Constructing social networks from text can be accomplished in several ways. An overall description of the process is shown in Figure 1. The simplest approach is to use links based upon the co-occurrence of entities in a document. This approach can be accomplished with simple string matches [8] or with full-scale NER [5]. This co-occurrence approach works reasonably well with certain genres of documents (e.g., newswire reports) in which two entities mentioned together are presumed likely to be related. This approach fails for long documents that cover a wide range of topics (e.g., a survey report). For the latter case, the co-occurrence approach can be refined to narrower parameters, such as "occurs in the same paragraph, sentence, or even subject topic" of a given report.

A second approach to extracting networks is to look for mentions of relationships in text [11]. For instance, in a document, Bob and Mary are referred to as brother and sister. This approach is compelling but has drawbacks. In many situations, relationships are implicit and not stated. The problem then becomes a process of inferring relations from text. Even with human annotators, making such inferences is a difficult task with high inter-annotator disagreement for some tasks [11].

A final issue in the extraction of entities from text is that of co-reference resolution. The problem arises because mentions of a name within and across documents

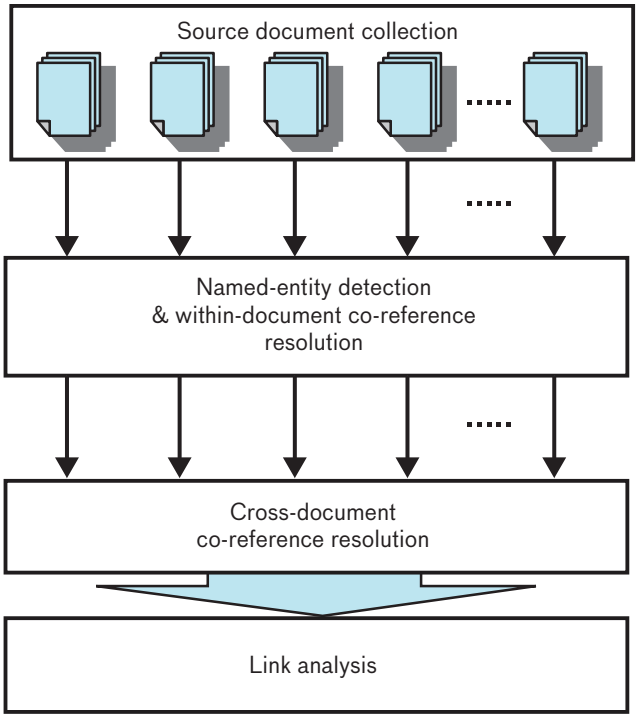

FIGURE 1. Information extraction from a corpus of documents for social network construction is accomplished by first performing named-entity detection and co-reference resolution within a document followed by co-reference resolution between documents. Networks can then be constructed using links of different types found between the entities.

vary—John, John Smith, Mr. Smith. Co-reference resolution combines all of these variants into one entity. However, a quick look at "John Smith" on Wikipedia shows that a name alone is not sufficient to disambiguate an entity. Co-reference resolution is difficult within documents, and across-document resolution is even harder.

Representation

Once a social network is extracted from the original data source, it must be stored in structured form so that automatic analysis, retrieval, and manipulation are possible. Multiple possible structures for the representation are mathematically equivalent. The main difference among them is that they arise in multiple fields. Two such representations are knowledge representation and graphs.

KNOWLEDGE REPRESENTATION

Extracted structured content from raw data can be encoded using standard knowledge representation methods and ontologies [12, 13]. For every input datum (e.g., text, speech, image), analysts produce a set of objects, attributes, and predicates conforming to an ontology that describes structured information in the document. An ontology based on standards for information extraction, primarily the Automated Content Extraction (ACE) protocol [11], is common.

A typical example extraction from a document might be *Member*(*Bob, KarateClub*) where *Bob* is an object of type *per* (a person) and *KarateClub* is an object of type *organization*. The statement *Member*(·,·) is a predicate and describes some relation between the entities.

An important point is that representation is usually limited to binary predicates, i.e., relationships of the form *Relation*(*entity, entity*). At first, this might appear to be a constraint. For instance, how does one represent multiple entities participating in a meeting or event? The key is to create a meeting entity, **M**, and then state all relations between all people, **P**, involved in the meeting and **M**, e.g., Participates(*Bob*, **M**), Participates(*Fred*, **M**), and so on.

Another property is that objects can have attributes. For instance, it is possible to extract *ATT_age*(*Bob*, 25). Attributes can be complex. For instance, one attribute of *Bob* could be the text document of his resume.

The knowledge representation approach is equivalent to a relational database model. In fact, the original work on relational databases by Codd [14] used a predicate model and corresponding "calculus" for manipulation. Each predicate corresponds to a table, and the entities in the relations are stored in the table. A typical example is shown in Figure 2.

Note that because our relations are binary, an alternate database structure is a triple store, which is designed to store and retrieve identities constructed from a set of three relationships. The triple in this case is (*predicate*, *val1*, *val2*). Triple stores have become popular lately in the Resource Description Framework (RDF) used by the World Wide Web consortium for the Semantic Web [15].

ATTRIBUTED GRAPHS

An alternate representation of social network data is to view the knowledge representation structure as a graph. The knowledge representation approach lends itself naturally to graph "conversion." Figure 3 shows the basic process through a restructuring of the data in Figure 2 as a graph. The basic process is as follows. Entities are converted to nodes in the graph. Note that the nodes

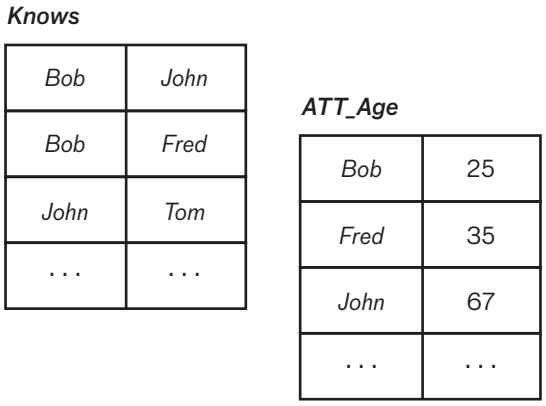

FIGURE 2. Knowledge representation in a relational database. Standard knowledge representation schemes usually involve binary predicates defining relationships between entities. This knowledge representation approach is equivalent to a relational database model as shown above for the predicates, *Knows* and *ATT_Age*.

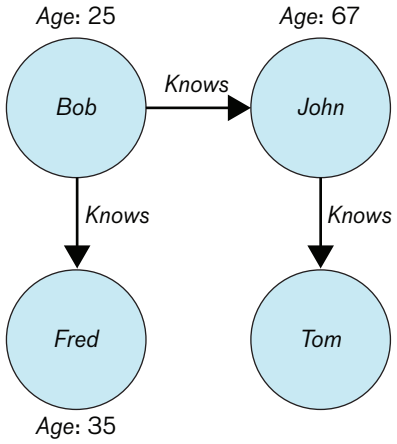

FIGURE 3. The directed edges of this attributed graph example show the unidirectional *Knows* relationship between *Bob* and both *John* and *Fred*, and the unidirectional relationship between *John* and *Tom*. This view of knowledge representation lends itself more easily to graph analysis problems and approaches.

can have different types—e.g., people, organizations, and events. This model deviates from the standard model of a graph in which the node set is one type. Edges in the graph correspond to relations between entities, so the relation *Knows*(*Bob*, *Fred*) is converted to a directed edge between nodes with the *Knows* attribute. Note that if we wanted a relation to be bidirectional [*Knows*(a, b) implies *Knows*(b, a)], an undirected edge could be used between nodes. The remaining process is to convert attributes of entities to attributes on nodes. Note that attributes on edges are also possible in both models. For instance, one may want to indicate the evidence for or the time of a relationship between two entities.

Although the mapping between graphs and the knowledge representation is straightforward once it is explained, both approaches inspire different perspectives and algorithms for viewing the data. For the graph approach, analysis of structure and communities in the data are natural questions. For the knowledge representation approach, storage, computation, manipulation, and potentially reasoning method become natural questions.

Community Detection

Many social networks exhibit community structure. Communities are groups of nodes that have high connectivity within a group and low connectivity across groups. Communities roughly correspond to organizations and groups in real social networks. For the purposes of this article, assume that the communities are disjoint, that is, membership in one community precludes membership in another.

In considering the problem of community detection for social networks, Lincoln Laboratory researchers applied multiple algorithms in the literature to the problem of community detection on the ISVG database. The goal was to partition a set of people into distinct violent groups. Because the ISVG database has labeled truth for people and organizations, the performance of multiple methods can be quantitatively measured. The Laboratory's research showed that, in contrast to comparisons in the literature using simulated graphs [16], no method was a clear winner in terms of performance.

ISVG Database

The Institute for the Study of Violent Groups is a research group that maintains a database of terrorist and criminal activity from open-source documents, including news articles, court documents, and police reports [10]. The database scope is worldwide and covers all known terrorist and extremist groups, as well as individuals and related funding entities. The original source documents are contained in the database along with more than 1500 carefully hand-annotated variable types and categories. These variables range from free text entries to categorical fields and continuous variables. Associations between groups,

individuals, and events are also included in the annotation. More than 100,000 incidents, nearly 30,000 individuals, and 3000 groups or organizations are covered in the database. The data are continually updated, but the version of the database used in the research reported here covered incidents up until April 2008.

Methods

Multiple methods for community detection have been proposed in the literature. Many of these methods are analogous to clustering methods with graph metrics. Rather than trying to be exhaustive, Lincoln Laboratory researchers selected three methods representative of standard approaches: Clauset/Newman/Moore (CNM) modularity optimization, spectral clustering, and Infomap.

MODULARITY OPTIMIZATION

Modularity optimization is a recent popular method for community detection [17, 18]. Modularity is an estimate of the "goodness" of a partition based on a comparison between the given graph and a random null model graph with the same expected degree distribution as the original graph [19]. A drawback of the standard modularity algorithms is that they do not scale well to large graphs. The method proposed by Clauset, Newman, and Moore [20] is a modularity-based algorithm that addresses this problem.

SPECTRAL CLUSTERING

Spectral methods for community detection rely upon normalized cuts for clustering [21]. A cut partitions a graph into separate parts by removing edges; see Figure 4 for an example. Spectral clustering partitions a graph into two subgraphs by using the best cut such that within-community connections are high and across-community connections are low. It can be shown that a relaxation of this discrete optimization problem is equivalent to examining the eigenvectors of the Laplacian of the graph. For this research, divisive clustering was used, recursively partitioning the graph into communities by "divide and conquer" methods.

INFOMAP

A graph can be converted to a Markov model in which a random walker on the nodes has a high probability of transitioning to within-community nodes and a low probability of transitioning outside of the community. The problem of finding the best cluster structure of a graph can be seen as the problem of optimally (losslessly) compressing the node sequence from the random walk process in an information theoretic sense [22]. The goal of Infomap is to arrive at a two-level description (lossless code) that exploits both the network's structure and the fact that a random walker is statistically likely to spend long periods of time within certain clusters of nodes. More specifically, the search is for a module partition $\mathbf{M}$ (i.e., set of cluster assignments) of N nodes into m clusters that minimizes the following expected description length of a single step in a random walk on the graph:

$$L(\mathbf{M}) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(P^i)$$

This equation comprises two terms: first is the entropy of the movement between clusters, and second is the entropy of movements within clusters, each of which is weighted respectively by the frequency with which it occurs in the particular partitioning. Here, $q_{\curvearrowright}$ is the probability that the random walk switches clusters on any given step, and $H(Q)$ is the entropy of the top-level clusters. Similarly, $H(P^i)$ is the entropy of the within-cluster movements and $p^i_{\circlearrowright}$ is the fraction of within-cluster movements that occur in cluster i. The specifics are detailed in Rosvall and Bergstrom [22].

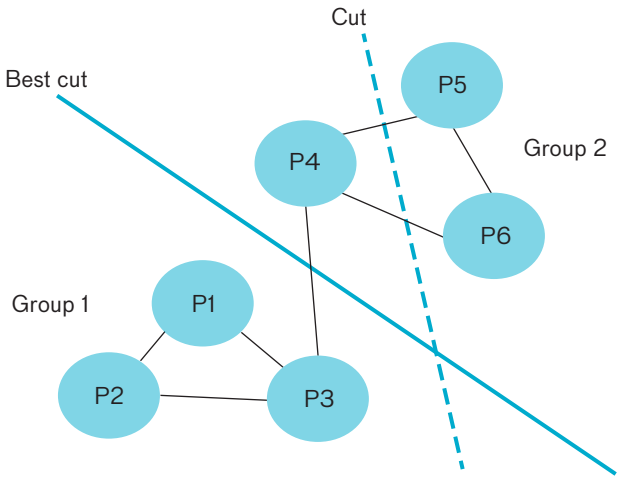

FIGURE 4. Spectral clustering of a graph relies on recursive binary partitions, or "cuts," of the graph. At a given stage, the algorithm chooses among all possible cuts (as illustrated by the dotted line) the "best" cut (shown by the solid line) that maximizes within-community connections and minimizes between-community connections of the resulting subgraphs.

Experiments

The first step in the experiments was to exploit the ISVG data to obtain a social network for analysis. Queries were designed in SQL to extract people and their associated mentions in documents. Then, a network of documents and individuals was constructed on the basis of document co-occurrence. The resulting graph is shown in Figure 5.

Community detection methods were then applied to the resulting graph. It is instructive to see an example. Figure 6 shows divisive spectral clustering. For the first step, the graph is split into two parts. Then, recursively the graph is split using a tree structure. The colors indicate the final communities.

Qualitatively, the communities found in Figure 6 corresponded to real violent groups. For example, the system was able to find Al Qaeda, Jemaah Islamiyaah, and the Abu Sayyaf violent groups with high precision. In general, the precision and recall of the algorithms can be quantitatively measured. For any two individuals, it was established if they were in the same violent group (or not) by using the truth tables from ISVG. Then, this fact was compared to the predicted membership obtained from the community-detection algorithm. A true positive (TP) occurs when both the groups and the communities are the same for the two individuals. A false positive (FP) occurs when the groups are not the same, but the individuals are placed in the same community. Finally, a false negative (FN) occurs when the groups are the same, but the community detection indicates they are not. The two measures of performance are then

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{and} \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} .$$

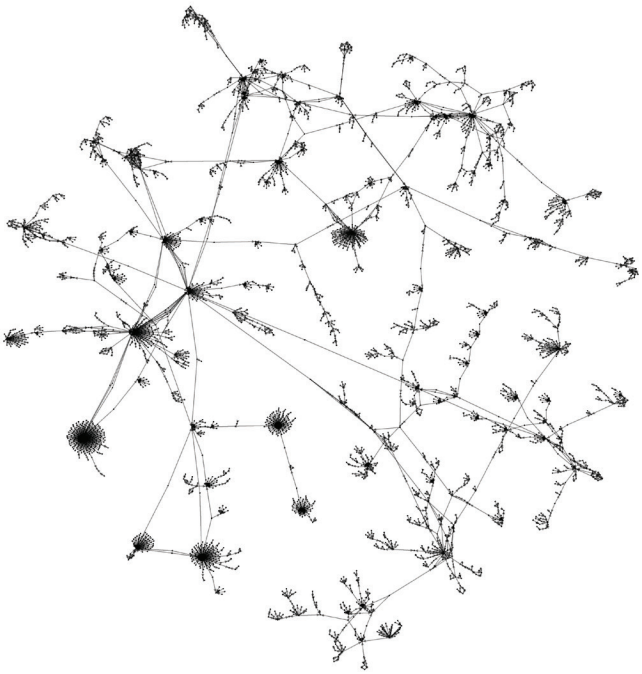

FIGURE 5. Largest connected component for ISVG individuals. This graph shows the document co-occurrence connections between individuals in the ISVG dataset. Highly connected individuals account for the small clusters of mass seen in the graph. Community-detection algorithms can help partition this graph with respect to this connectivity, giving a summarized view into the data.

The quantitative comparison of the various algorithms is shown in Figure 7 using a precision/recall curve. The figure shows that both the CNM and Infomap algorithms produce high precision. The spectral clustering algorithm has a threshold that allows a wide variation in the trade-off of precision versus recall. In general, the trade-off is due to the community (cluster) size. The algorithms can either produce small clusters

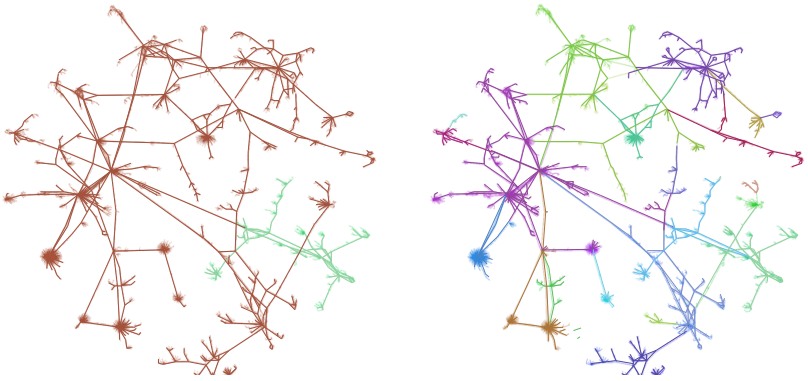

FIGURE 6. Divisive spectral clustering on the ISVG social network graph. The left figure is the first split; colors indicate the split into two groups. The right figure shows the final partitioning into groups after multiple iterations.

that are highly accurate or larger clusters that are less accurate but have better recall. Overall, users of these algorithms will have to determine what operating point is best fitted to their application.

Community Dynamics

As discussed in the previous section, community detection is a fundamental component of network analysis for sensor systems and is an enabling technology for higher-level analytical applications such as behavior analysis and prediction, and identity and pattern-of-life analysis. In both commercial industry and academia, significant progress has been made on problems related to the analysis of community structure; however, traditional work in social networks has focused on static situations (i.e., classical social network analysis) or dynamics in a large-scale sense (e.g., disease propagation).

As the availability of large, dynamic datasets continues to grow, so will the value of automatic approaches that leverage *temporal* aspects of social network analysis. Dynamic analysis of social networks is a nascent field that has great potential for research and development, as well as for underpinning higher-level analytic applications.

Overview of Dynamic Social Network Analysis

Analysis of time-varying social network data is an area of growing interest in the research community. Dynamic social network analysis seeks to analyze the behavior of social networks over time [23], detecting recurring patterns [24], community structure (either formation or dissolution) [25. 26], and typical [27, 28] and anomalous [29] behavior.

Previous studies of group-based community analysis have looked into more general analyses of coarse-level social dynamics. Hierarchical Bayesian topic models [30], hidden Markov models [1], and eigenmodeling [31] have been used for the discovery of individual and group routines in the reality mining data at the "work," "home," and "elsewhere" granularity levels. Eagle, Pentland, and Lazer [32] used relational dynamics in the form of spatial proximity and phone-calling data to infer both friendship and job satisfaction. Call data was also used by Reades et al. [33] to cluster aggregate activity in different regions of metropolitan Rome.

Lincoln Laboratory's work to elaborate on research in this area studied how sociographic data can be used for the analysis and prediction of individual and group behavior in dynamic real-world situations. Specifically, the Laboratory's researchers explored organizational structure and patterns of life in dynamic social networks [2] through analysis of the highly instrumented reality mining dataset gathered at the Human Dynamics Laboratory (HDL) at the MIT Media Lab [32].

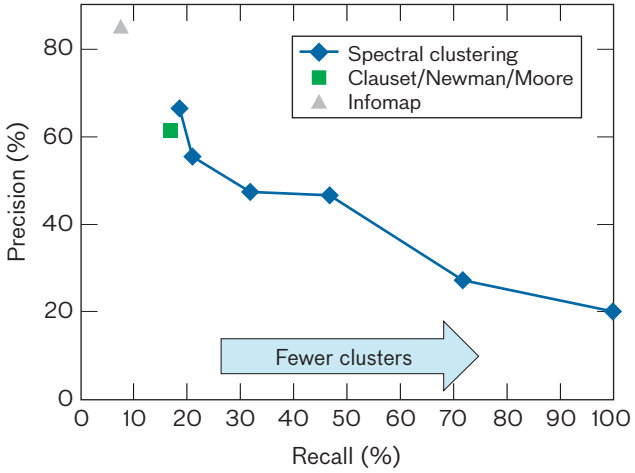

FIGURE 7. Precision and recall for various algorithms and thresholds on the ISVG social network graph. As can be seen from the figure, there is a large range of operating points within which community-detection algorithms can operate. End-users of these algorithms will have to determine the trade-off between precision and recall for their applications of interest.

Reality Mining Dataset

The Reality Mining Project was conducted from 2004 to 2005 by the HDL. The study followed 94 subjects who used mobile phones preinstalled with several pieces of software that recorded and sent to researchers data about call logs, Bluetooth devices in proximity of approximately 5 meters, location in the form of cell tower identifications, application usage, and phone status. Over the course of nine months, these measurements were used to observe the subjects, who were students and faculty from two programs within MIT. Also collected from each individual were self-reported relational data that were responses to subjects being asked about their proximity to, and friendship with, others.

The subjects were studied between September 2004 and June 2005. For the Lincoln Laboratory experiment, 94 subjects who had completed the survey conducted in January 2005 provided the data for analysis. Of these 94

subjects, 68 were colleagues working in the same building on campus (90% graduate students, 10% staff), and the remaining 26 subjects were incoming students at the Institute's business school (Sloan School). The subjects volunteered to become part of the experiment in exchange for the use of a high-end smartphone for the duration of the study. Interested readers are referred to [1] for a more detailed description of the Reality Mining Project.

The Reality Mining Project's data were obtained from the MIT HDL in anonymized form. All personal data such as phone numbers were one-way hashed (using the MD5 message-digest algorithm), generating unique identities for the analysis. MIT HDL found that, although subjects were initially concerned about the privacy implications, less than 5% of the subjects ever disabled the logging software throughout the nine-month study. Data were reformatted into a MySQL database to enable easier querying and anomalous information filtering.

Inferring Dynamic Community Behavior

Lincoln Laboratory researchers sought a general approach for dynamic social network analysis that would allow for fast, high-level summaries of, and insight into, group dynamics in real-world social networks. To this end, we investigated the use of multilinear semantic indexing (MLSI) [29, 34] in the context of dynamic social networks. Multimodal co-clustering tools, based on tensor modeling and analysis, can be successfully used to provide fast, high-order summarizations of community structure and behavior.

MULTILINEAR SEMANTIC INDEXING

Multilinear semantic indexing is a generalization of traditional latent semantic indexing (LSI). To see this connection, consider the use-case of text document clustering. Traditional LSI relies on the rank-k singular value decomposition (SVD) of the term-document matrix, a matrix of (weighted) term frequency (rows) as a function of corpus document (columns). This decomposition creates topic-term and topic-document clustering through the respective sets of k left and right singular vectors. These vectors are called *aspect profiles*. Additionally, large singular values weight the strength of correlation between pairs of document/term aspect profiles, serving to *co-cluster* documents and terms within the corpus.

MLSI generalizes this idea to create *multimodal co-clustering* from higher-order tensor representations of the data: term-document-author tensors for the text document clustering example. Through high-order SVD (HOSVD), MLSI produces M multimodal aspect profiles (topic-term, topic-document, topic-author) as well as multimodal co-clustering between M-tuples (ordered lists) of aspect profiles through the multilinear equivalent of singular values. HOSVD can be accomplished through algorithms that compute a *Tucker decomposition* of the input tensor [34].

Traditional SVD algorithms seek to model input matrices as a weighted sum of rank-1 outer products between pairs of vectors (document/term aspect profiles in the text processing example). Analogously, HOSVD via Tucker decomposition seeks to explain the data as a weighted sum of rank-1 tensors, which are the result of M-way outer products between M-tuples of vectors. The corresponding weights for these M-tuples are stored in the so-called *core* tensor.

Accordingly, larger absolute values in the core tensor mean a larger contribution to the final reconstruction, which therefore implies the relative importance of the M-tuple of subspace vectors (as a group) in approximating the input tensor. In this way, the vectors in these M-tuples can be considered strongly correlated. Values in the core tensor highlight these correlations and, as a result, provide meaningful co-clustering between M-tuples of vectors from each modality's projection space. Thus, the core tensor serves an analogous role to the singular values in traditional LSI. The columns of the projection matrices correspond to a multimodal version of traditional LSI aspect profiles.

MLSI FOR SOCIAL INTERACTION DATA

In the case of time-varying social network interactions (where time is considered the third index in an order-3 data tensor in which indices 1 and 2 are relationships between actors), the Tucker decomposition may be more easily interpreted through *time-profile-specific subnetworks*. Specifically, this means reinterpreting the set of 3-tuples of aspect profiles created by MLSI as a tuple of a network matrix (created by the outer product between the two actor relationship vectors) and a correlated time profile (the vector corresponding to the temporal aspect).

The network matrix can be interpreted as the adjacency matrix of the social interactions correlating most strongly to a given time profile. When time is the third

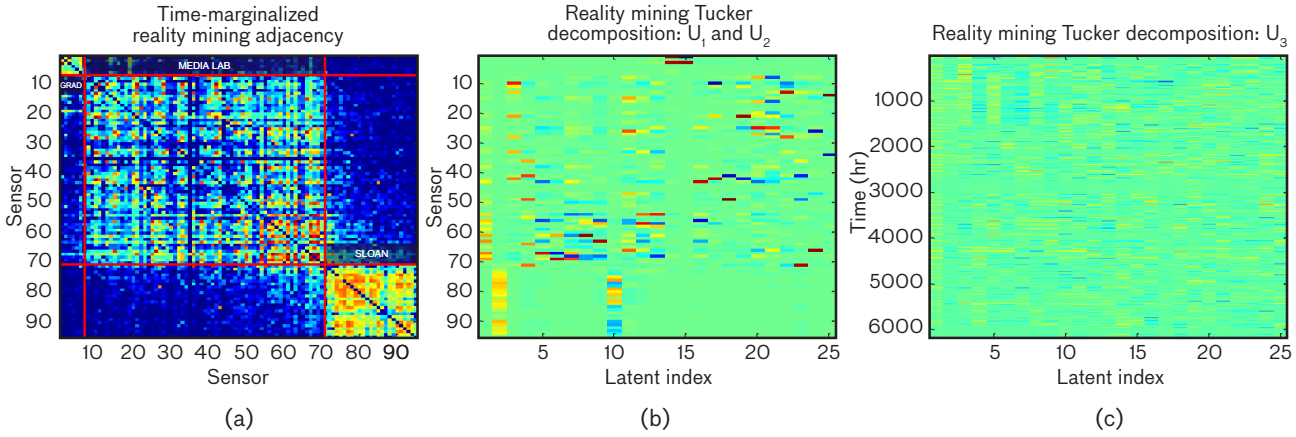

FIGURE 8. Marginal adjacency and Tucker decomposition of reality mining data. Distinct community structure is observed in the time-marginalized adjacency matrix of the reality mining dataset (a). The three delineated blocks correspond to "General Graduate," "MIT Media Lab," and "Sloan" student clusters respectively. The Tucker decomposition is shown in (b) and (c). Because of the symmetric property of the adjacency matrix, indices 1 and 2 produce identical projection matrices; therefore, only one copy is included here for clarity. It is easy to see clear community structures in the social network profiles, most notably the distinction between Media Lab and Sloan students.

index in the data tensor, this interpretation may be a more natural way to interpret a Tucker decomposition as opposed to traditional interpretations of MLSI in which the list of most active participants from each correlating pair (or tuple) of indices is returned by the system in list form.

To gain insight on the ability of this approach to summarize group behavior from dynamic social network data, close-range social interactions were analyzed from the Reality Mining Project's corpus.

Experimental Protocol

For these experiments, the researchers used Bluetooth proximity information from periodic scans of nearby devices from each of the 94 participants in the reality mining study. The proximity information resulted in more dense interaction networks than call data when small time increments were considered.

The number of detections between study participants per hour was used as the social interaction feature. Restricting the time range to the academic year (1 September 2004 to 15 May 2005) and removing participants lacking Bluetooth data resulted in a mode-3 data tensor of size $95 \times 95 \times 6168$. That is, for input tensor $\mathbf{X}$, $(x_{i,j,k})$ corresponds to the number of times participant i was detected by participant j at hour k.

The values in the input tensor were nominalized by $\log(1 + x_{i,j,k}) \forall x_{i,j,k}$ to prevent large values from dominating the tensor decomposition. For ease of interpretability, the indices of the input tensor were reordered by using spectral clustering (as described earlier). To do so, the second-largest eigenvector of the normalized graph-Laplacian of the time-marginalized social network was calculated (Figure 8a). The values in this eigenvector were then sorted, and the resulting reordering of indices was used to reorder indices in the input tensor.

For each dataset, MLSI was performed via a rank-(25, 25, 25) Tucker decomposition of the preprocessed input tensor, keeping the original time resolution on samples. The size of the decomposition was chosen heuristically to represent a reasonable number of profiles users could sort through when interpreting the results. Both the size of the decomposition and the time resolution of the input tensor are open issues when this approach is used, and a discussion of these issues is deferred until the experiment sections. Results were generally evaluated qualitatively.

Results

The first two of the social profile vectors in Figure 8b correspond loosely to the two main communities seen in the Reality Mining Project's data, namely the Media Lab and Sloan participants. Subsequent profile vectors corresponded to the larger, more active Media Lab participants. In general, researchers observed reasonable community clustering and co-clustering where it existed.

Temporally, the expected gaps in the time profiles corresponded to the Thanksgiving holiday, 25–29 November [35], and winter break, 18 December–2 January* [35]. The disbanding of the Sloan School community as a whole was detected after the fall semester, and was easily observable in the sharp drop-off of signal energy in time profile 2 as seen in Figure 9b. Additionally, spectral analysis showed that the time profiles exhibited behavior corresponding to daily and weekly routines: the two largest spectral components (on average) are 1/24 hours and 1/163 hours (1/6.8 days). This specific result was also observed by Eagle and Pentland [1]. The time profiles seen in Figure 8c are more clearly interpreted in the context of their correlated subnetworks, which are shown in Figure 9a.

It is instructive to view the results of the MLSI analysis with respect to the two main communities represented among the participants in the study: the Media Lab and Sloan graduate students.

MEDIA LAB COMMUNITY

The first time-profile-specific subnetwork, Figure 9a, was the most relevant subnetwork for the Media Lab community, and in general, a compact first-order summary of the data. The time profile exhibited a fairly uniform structure across all time instances, with general periodic features consistent with work- and school-related activities throughout the academic year. Most noticeably, the gap in the time profile corresponded to the break between fall and spring semesters. The subnetwork showed that the majority of the social interactions occurred among the larger Media Lab community (with a relatively smaller proportion occurring in the Sloan community). This information differed from that conveyed in the average social network derived from marginalizing interactions over time (seen in Figure 8a). When time information was removed, it appeared the Sloan community was equally active as the Media Lab community during this time period. In reality, as this subnetwork and subsequent subnetworks arising from tensor analysis showed, this was not the case.

SLOAN GRADUATE STUDENTS

The most interesting results of the MLSI analysis of the Reality Mining Project's data were subnetworks (and associated time profiles) correlating with the Sloan graduate-student community. As can be seen in Figure 9b, both the Sloan student subnetwork and the corresponding time profile were detected cleanly. From the time profile, this subnetwork appeared strongly only in the fall semester.

Figure 10b lists the most prominent time stamps from each peak cluster (sorted chronologically) from this time profile. Interestingly, the strongest periodic behavior can be observed on Tuesdays and Thursdays at 11:00 a.m. from the middle of October to the beginning of December. If one zooms into the time profile, local peaks can be observed consistent with this result. Figure 10a shows a three-week window between 24 October and 22 November 2004. The red bars on the time axis delineate weekday from weekend. At this finer time resolution, an additional weekly subpeak associated with Thursdays at 4:00 p.m. can be observed. These same spikes on Tuesdays and Thursdays at 11:00 a.m. and Thursdays at 4:00 p.m. appeared locally throughout the profile, not just in weeks associated with the global peaks.

Because the spikes were clearly periodic, persisted through the fall semester, and then disappeared in the spring, the evidence suggested this behavior could be due to course-attending activity. Of all Sloan course offerings in fall 2004 [36], only two courses were consistent with this behavior. Of these, only the first-year core course, 15.515, Financial Accounting [37], fit all the observed evidence. As seen in Figure 10c, 15.515 had three lecture sessions that met Tuesdays and Thursdays from 10:30 a.m. to 12:00 p.m., and two recitation sessions that met Thursdays at 4:00 p.m. Therefore, one could make the uncertain, yet probable, claim that the behavior observed in this time profile corresponded to Sloan students attending 15.515, Financial Accounting.

This MSLI approach is ideally suited for scenarios in which actors co-cluster cleanly in space and time. Given the appropriate context, MSLI analysis allowed researchers to establish (with some degree of uncertainty) a causal link between an observed behavior and a generating event, a result that was not readily apparent from the input data.

Tensor-based analysis tools such as MLSI are fast, first-order tools that could allow users to refocus both their attention and the attention of more resource-inten-

* Last day before the Independent Activities Period during the January intersession.

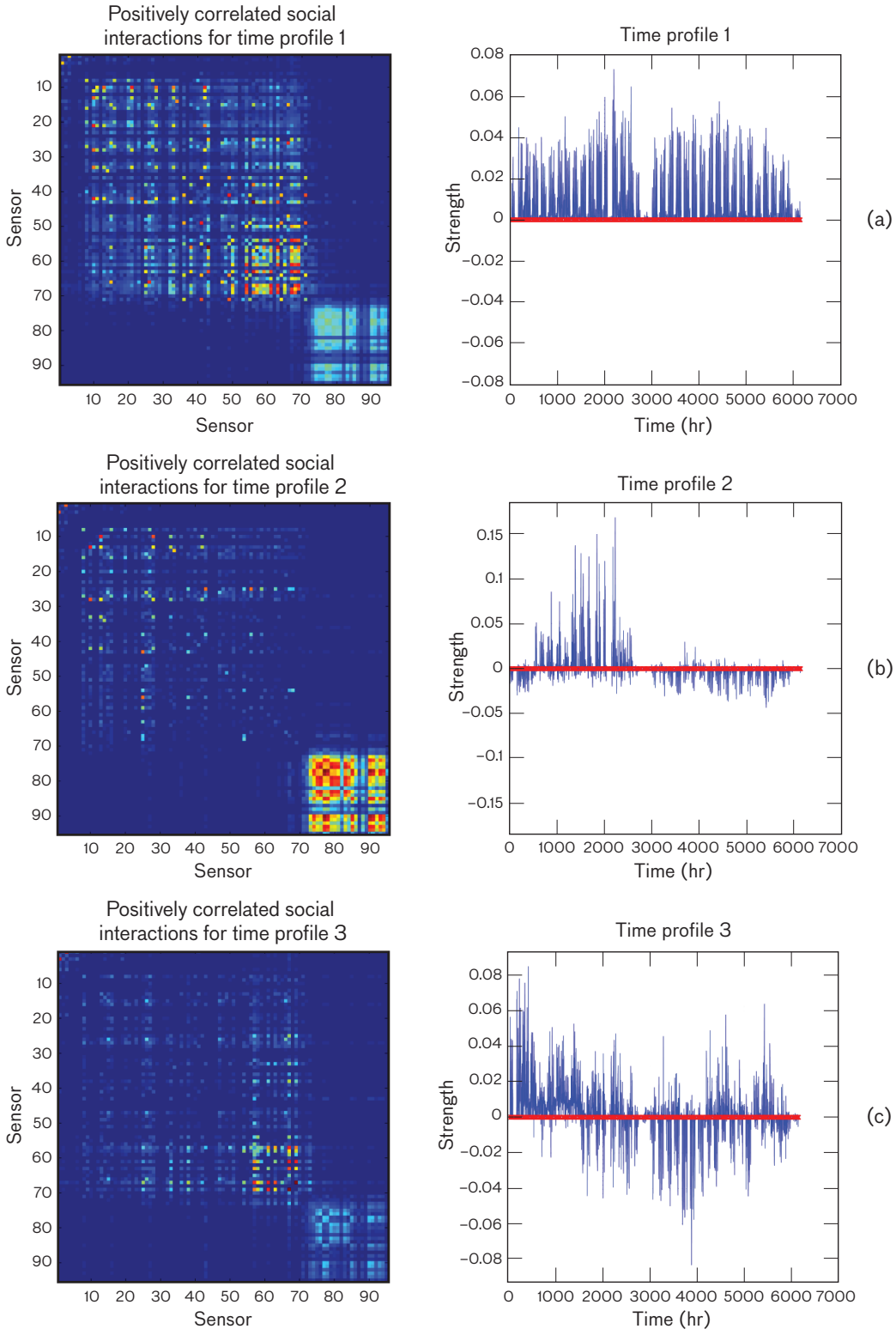

FIGURE 9. Time-profile-specific subnetworks for Reality Mining Project dataset. Subnetworks and their correlated time profiles are shown in pairs for time profiles 1 (a), 2 (b), and 3 (c) respectively. Positive and negative values in the time profile indicate positive and negative correlations, respectively, with the associated subnetwork. This representation provides a richer interpretation for communities and their behavior as compared to the results shown in Figure 8.

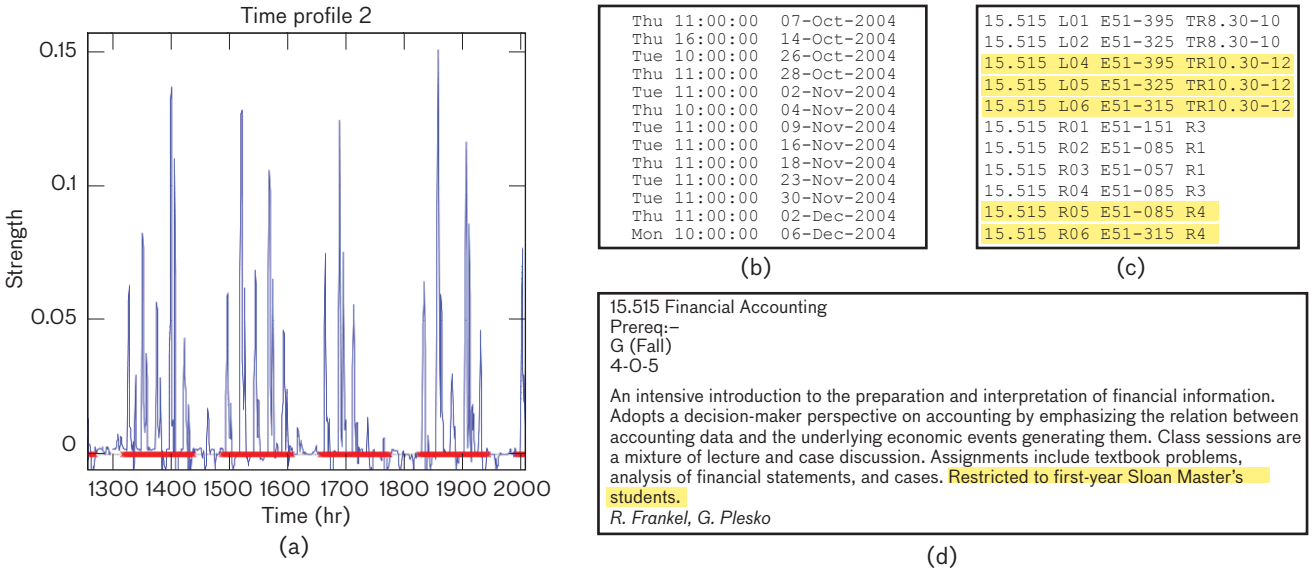

FIGURE 10. Possible course-attending behavior. The time stamps corresponding to the strongest peaks in time profile 2 (b) show strong periodic behavior on Tuesdays and Thursdays at 11:00 a.m. Of the two first-year core courses in the fall 2004 course catalog consistent with these times, only 15.515, Financial Accounting (c), looks consistent with the data. Figure 10a shows local spikes occurring consistently on Tuesdays and Thursdays at 11:00 a.m., as well as minor spikes on Thursdays at 4:00 p.m. (one of two recitation sessions meeting at that time). This behavior persists throughout the fall semester and disappears in the spring. From this analysis, researchers hypothesized the subnetwork corresponding to this time profile could be those Sloan students attending the first-year core course, 15.515, Financial Accounting (d).

sive analytics, such as relational learning, further down the processing chain.

Relational Learning

Prior sections focused on constructing social networks, finding communities, and analyzing dynamic patterns of activity. This section considers the problem of inference on graphs. Given graphs with a rich attribute structure and a statistically large sample, it is possible to perform statistical relational learning on them. These methods learn models that relate attributes in a graph neighborhood of a given individual. These Bayesian graphical models can be used to impute missing values, perform prediction, and interpret classification results.

Lincoln Laboratory used statistical relational learning algorithms to predict leadership roles of individuals in a group on the basis of patterns of activity, communication, and individual attributes [38]. By using labeled training data, analysts applied supervised learning to build a model that described the structures and patterns of leadership roles. The relational model returned a probability that a particular person is in a leadership role, given a graphical representation of the individual's activities and attributes. A held-out test dataset was used for evaluation, and receiver operator characteristic (ROC) curves for correct prediction of leadership were presented. A more complex model was applied to give improved performance in a more realistic "data-poor" test condition. Such features can be important components of an overall automatic threat detection system. In such a system, automatic identification of individual roles and activities from basic features can help infer the intent of groups and individuals through higher-level pattern recognition and social network analysis.

Graphical Schema

For the research on predicting leadership roles, Laboratory analysts used a subset of the overall ISVG database schema that contained most of the categorical fields and continuous variables available in the database. A continuous variable may consist of a date, age, number of casualties, or other such variables represented by a single number. Categorical fields generally represent the type of a particular object in the database and may include incident types (e.g., bombing, armed assault, kidnapping) or specific information about weapons or bombs used in an attack.

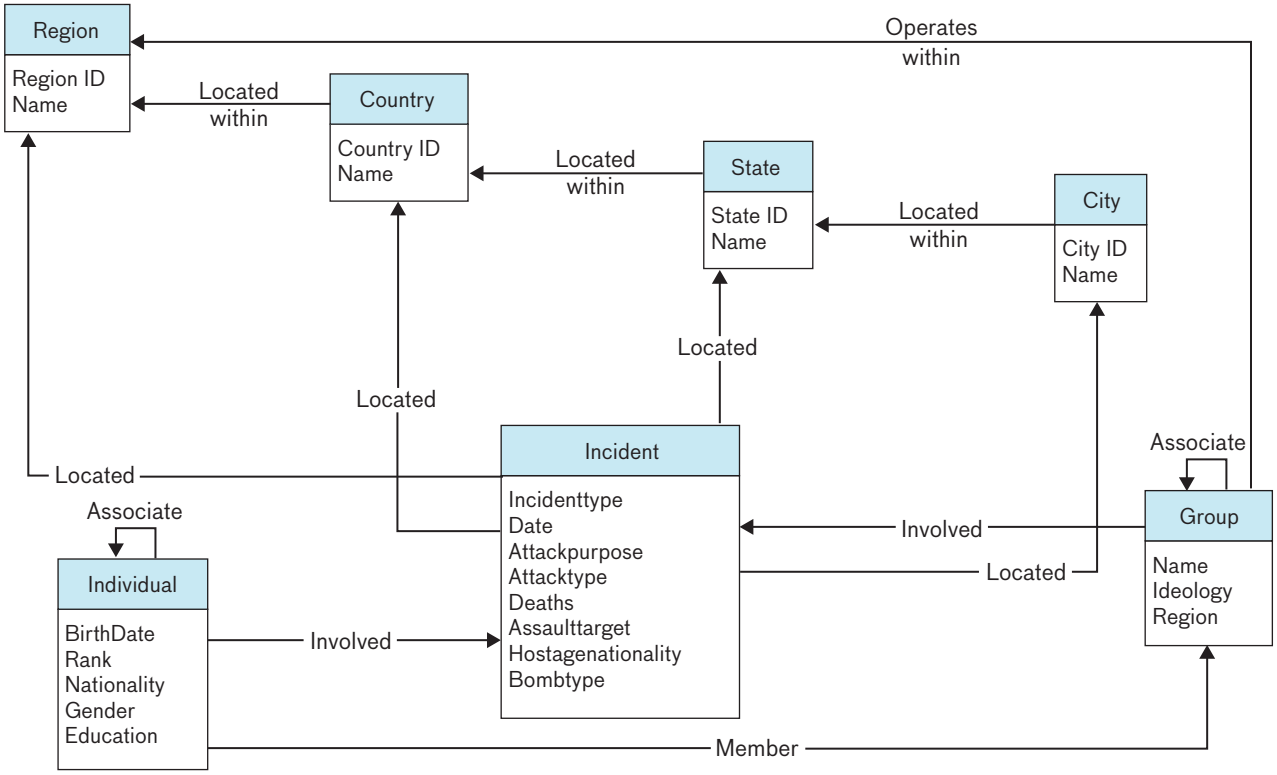

FIGURE 11. Sample ISVG database schema. Each node in the schema represents an entity type and the available attributes for that entity. Entities can be linked by relationships of different types. In total, there are 9 node, 90 attribute, and 11 link types.

A sample of the schema is represented graphically in Figure 11. Each node represents an object type, and the text below the object type represents the available attribute fields for each object. For example, *individuals* have a "birth date," "nationality," "gender," and other attributes. Objects are linked via a specific link type indicated by the text on the line. At the center of the schema is the incident, and groups and individuals connect to particular incidents via their involvement. The entire dataset consists of 9 different node types, 90 different attribute types, and 11 link types. The actual instantiation of the graphical database contains more than 180,000 nodes with more than 2 million attributes spread across the 90 attribute types. Nodes are connected by more than 1 million links.

Graphical Query

Once the database was represented as a graph with nodes, edges, and attributes, QGraph software was used to pull selected subgraphs from the larger database for analysis. QGraph is a graphical query language designed for querying large relational datasets, such as social networks [39]. Queries are specified visually by drawing the structure of the desired matches and adding annotations to that structure to further refine the query. Matches are returned as subgraphs, which are small subsections of the overall data containing only the desired structure.

Methods and Technical Solutions

Classification uses statistical methods to predict the status of an (unknown) characteristic, or feature, of a particular entity given a set of observed characteristics also on the entity. Most classification algorithms assume data are independent and identically distributed (i.i.d.). However, because of the connections inherent in social, technological, and communication networks, data arising from these sources do not meet either of these conditions. For example, in criminal networks, known associates of convicted criminals are likely to be criminals as well (nonindependent), and some criminals have many more associates than others (heterogeneous). Furthermore, network data often exhibit autocorrelation among class labels of related instances [40]. The concept of autocorrelation, sometimes called homophily, is best summarized by the phrase "birds of a feather flock together," indicating that

individuals with similar characteristics tend to be related. Failure to account for nonindependence and heterogeneity in network data can lead to biases in learned models when traditional approaches are used for classification [41, 42]. While traditional classification algorithms can incorporate relational features, an exhaustive aggregation of relational features becomes less efficient as the dataset becomes large. Even with the incorporation of relational features, the standard classification approach still makes predictions for each instance that are independent, making collective classification more difficult.

OVERVIEW OF STATISTICAL RELATIONAL LEARNING

Statistical relational learning (SRL) is a subdiscipline of the machine learning and data mining communities [43]. As its name implies, the focus of SRL is extending traditional machine learning and data mining algorithms for use with data stored in multiple relational tables, as typically occurs in a relational database, such as MySQL or Oracle. This storage model permits analysis of data that are nonindependent and heterogeneous, such as social network data. The primary focus of this work is classification in social network data. For example, in criminal networks, analysts are often interested in predicting a binary variable indicating whether a particular individual will commit a crime in the near future. The true value of this variable is generally unknown at the time of analysis; however, there are a number of observable features that are predictors, such as whether the individual or a closely related individual has committed a crime in the past, has recently filed bankruptcy, or has lost a job. Tools developed in the SRL community extend the traditional classification paradigm to include features on both the individual in question and features on individuals related through social or organizational ties.

In addition to using features on related individuals, social network data also provide the opportunity for collective classification. Collective classification is possible when many individual class labels (e.g., future crime status) are unknown but are connected via social or organizational ties. These relations among individuals permit the predictions about one individual to propagate to predictions about related individuals. Collective approaches, which infer the value of all unknown labels simultaneously, have been shown to yield higher accuracies than noncollective models, particularly when the labels of related instances exhibit autocorrelation [44]. Thus, collective classification is widely studied within the field of SRL [45]. Two specific SRL techniques for classification in relational data are presented here: the relational probability tree and relational dependency network.

RELATIONAL PROBABILITY TREES

The relational probability tree (RPT) is a probability-estimation tree for classification in relational domains [46]. A probability-estimation tree is a conditional model similar to a classification tree; however, the leaves contain a probability distribution rather than a class label assignment [47]. To account for nonindependence in network data, the RPT is designed to use both intrinsic features on the target individual and relational features on related individuals. However, because of heterogeneity in the data, the number of relational features can vary from individual to individual. To account for possible heterogeneity, the RPT automatically constructs features by searching over possible aggregations of the training data. The RPT applies standard aggregations—e.g., COUNT, AVERAGE, MODE—to dynamically flatten the data before selecting features to be included in the model. To find the best feature, the RPT searches over values and thresholds for each aggregator. For example, to aggregate over criminal activity of an individual, an appropriate feature might be [COUNT(CriminalActivity.type=Larceny) > 1], where the type and number are determined by the algorithm. The RPT has successfully predicted high-risk behavior in the securities industry in the United States by using the social network among individuals in the industry [48, 49].

RELATIONAL DEPENDENCY NETWORKS

The relational dependency network (RDN) is a joint relational model for performing collective classification [50]. An RDN is a pseudolikelihood model consisting of a collection of conditional probability distributions (CPD) that have been learned independently from data. The CPDs used in a dependency network are often represented by probability-estimation trees, although any conditional model suffices [51]. Lincoln Laboratory's work used an RDN consisting of a set of individually learned RPTs for each attribute that were combined into a single, joint model of relational data. Inference (prediction) in the RDN was accomplished by using Gibbs sampling, a technique that relies on repeated sampling from conditional

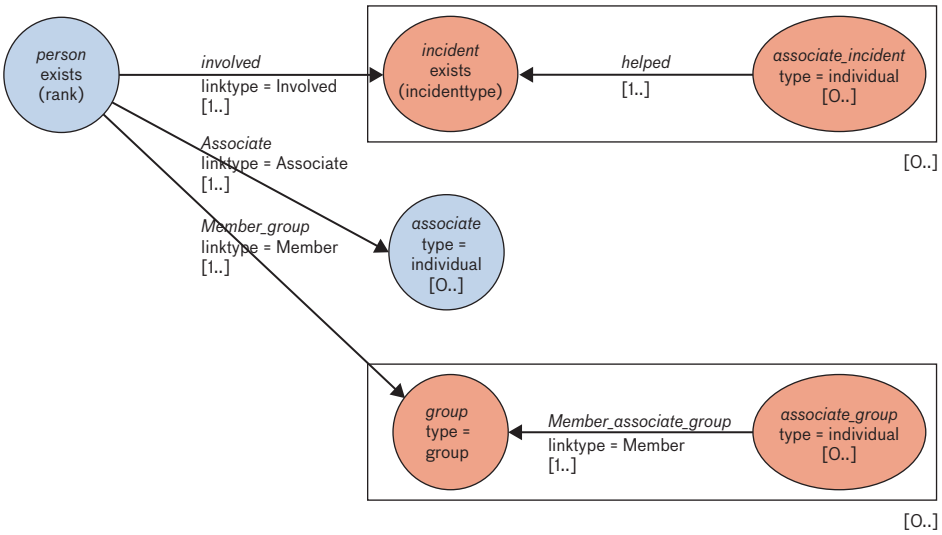

FIGURE 12. Graphical query for leadership. The query looks for all individuals where the rank is specified (*person* exists), individuals associated with that person (*associate*), incidents and people associated with those incidents (upper box), and groups and groups associated with those groups (lower box). Nodes indicate *entities* (people, groups, and incidents) to look for in the graph. The desired *attributes* (e.g., "helped") are specified on the nodes and edges. Subqueries are indicated using the plate (rectangle). The notation [*n*..] indicates that there should be at least *n* cases of a link or node.

distributions [52]. The RDN can represent autocorrelation relationships and was the first joint model that permitted the learning of autocorrelation relationships from data. Collective classification was performed via inference using multiple iterations of Gibbs sampling whenever relational features were included in the learned trees.

Empirical Evaluation

Experiments were performed using the ISVG relational database. Each of these experiments required a labeled set of subgraphs for training of the relational models and another, nonoverlapping set, for evaluation. Using the QGRAPH software, Lincoln Laboratory researchers constructed appropriate queries to return these subgraphs and randomly divide them into training and test sets using fourfold cross validation. After the learned model was applied to the evaluation set on one fold of the randomly selected data, the results were presented in ROC performance curves.

The target application was the prediction of leadership attributes of individuals within a group. The ISVG data contains a rich set of individual roles. For the Laboratory's research, these roles were binned into categories pertaining to leadership (e.g., field commander, cell leader, spiritual leader) and nonleadership (e.g., group member, aide, activist), resulting in a binary classification. Different learning techniques were applied, highlighting the differences in data-rich versus data-poor operational conditions.

LEADERSHIP ROLE PREDICTION

The ISVG database contains 3854 individuals with labeled roles. These roles were binned into leadership and nonleadership categories, and QGRAPH was used to extract relevant subgraphs for training and testing. There were 2890 randomly selected subgraphs for training and 964 for evaluation. The query is shown in Figure 12. The query looked for persons with a leadership attribute and all of their associates, including those related through the same group or incident as well as the groups and incidents themselves. The query in Figure 12 contains two subqueries (rectangular boxes) that look for zero or more incidents and groups, along with individuals involved in the incident or members of the group. In this way, the total number of related individuals was expanded beyond what the ISVG annotator labeled in the "Associate" link. Additionally, the incident attributes and group type could indicate an organizational structure to help predict leadership.

The first experiment assumed a data-rich condition in which full information about neighboring associates is known (e.g., age, education level, nationality, and so on); see Figure 13. All of this information was used in the RPT model

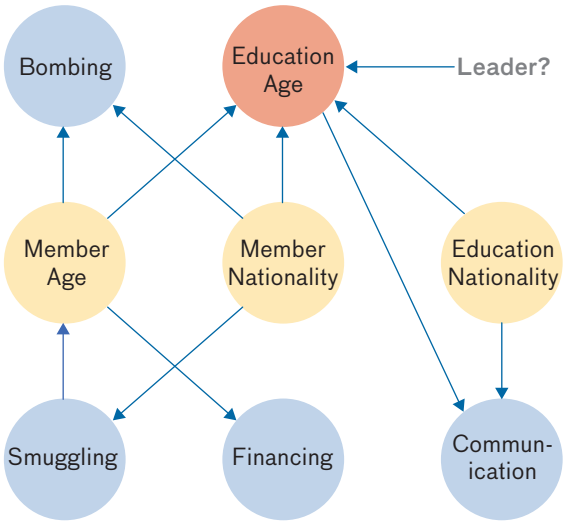

FIGURE 13. Ideal case of relational classification; many neighboring attributes are known.

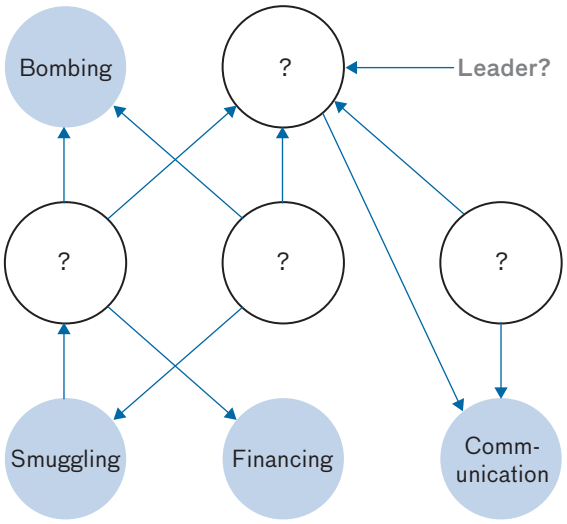

FIGURE 14. Realistic data condition in which attributes of associates are unknown.

to predict whether or not the node under consideration was in a leadership role. Under a more realistic assumption, this information may not be known; see Figure 14. In this case, a pattern of activity and communication may be observed, but we could determine very little about the actors involved and wished to determine who the leader was.

The results for both of these RPT models on the held-out evaluation dataset are shown in Figure 15 and Table 1. Figure 15 is a ROC curve; the probability of detection is plotted on the y-axis and the probability of false alarm is plotted on the x-axis. In this instance, a correct detection occurs when the system correctly predicts a leadership role for an individual. A false alarm occurs when the system predicts a positive leadership label for an individual who is not in a leadership role. Table 1 shows the area under each ROC curve. In Figure 15, the dotted line labeled "Ideal data (RPT)" represents the query results from Figure 13, in which all information about associates is known. The "ideal data" curve represents an upper bound on performance if all information is known about the individuals, including the leadership characteristics of the associates. The dashed line labeled "Realistic data (RPT)" represents the more realistic condition in which specific attributes of associates are hidden from the RPT model. The former represents the data-rich case and gives the best performance. In more realistic data conditions, the performance is significantly worse. The use of relational dependency networks can be used to improve performance in the latter, data-poor, condition.

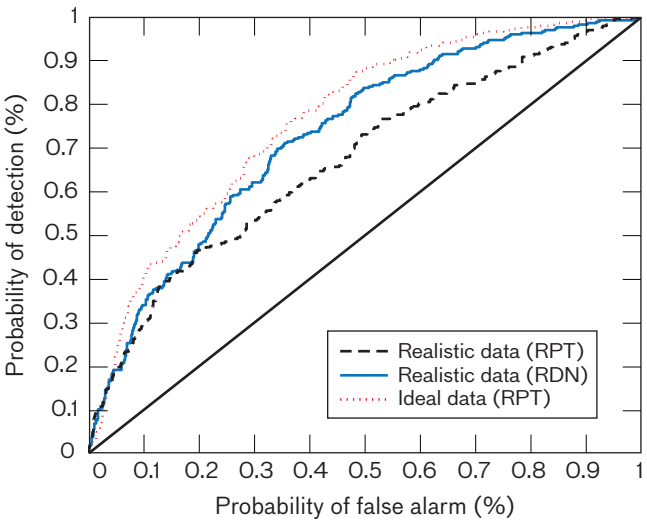

FIGURE 15. Performance of social network leadership prediction models.

Table 1. Area Under ROC Curve Performance of Social Network Leadership Prediction Models

CONDITION	AREA UNDER CURVE
RPT realistic data	0.6725
RDN realistic data	0.7314
RPT ideal data (upper bound)	0.7672

COLLECTIVE CLASSIFICATION

Relational dependency networks can perform collective classification when several attributes across selected variables are estimated simultaneously on the basis of a joint probability model. In this experiment, individual conditional models (RPTs) were built for each distinguishing attribute of the associate. These included the following variables: *leader*, *status*, *education*, *nationality*, and *race*. Multiple Gibbs sampling iterations were used to approximate the joint distribution. The results are shown in the solid line labeled "Realistic data (RDN)" in Figure 15. The RDN results are almost as good as the upper bound data-rich condition of the original RPT model. This result is promising as it becomes possible to predict leadership roles with some degree of accuracy for situations in which very little specific information is known about the individual actors.

Additional insight into the data can be learned from the relational dependency network diagram in Figure 16. The interpretation of the RDN is that of a relational extension of a "dependency network," a type of model in which arcs between variables indicate strict dependence rather than the more complex encoding of independence that the arcs in a Bayes net indicate. The large colored boxes (plates) represent entities. White circles indicate variables on those entities and arrows indicate dependence. The structure has been learned automatically from data, with RPTs underlying the individual attributes in the RDN. The figure shows three kinds of dependencies:

1. Dependence between variables on the same entity
   - The region of an incident and the incident type are dependent on each other.
2. Dependence between variables on different entities
   - The type of an organization depends on the country of incidents to which the organization has been tied.
3. Autocorrelation (special case of 2)—dependence on the same attributes across different entities
   - Incident type is autocorrelated through groups (i.e., groups tend to be involved with the same types of incidents).
   - CDOC (court document) charge and CDOC convictions of individuals are both autocorrelated through incidents. Individuals involved in the same incidents tend to have the same charges and conviction status.

In Figure 16, autocorrelation through an intermediate entity is indicated by a colored box on the self-loop. Individuals are autocorrelated through incidents, indicating the individuals with similar attributes tend to be involved in the same incidents. Self-loops with no box indicate direct linkage via known associates. Individuals with the same status (incarcerated, at large, etc.) tend to be directly associated.

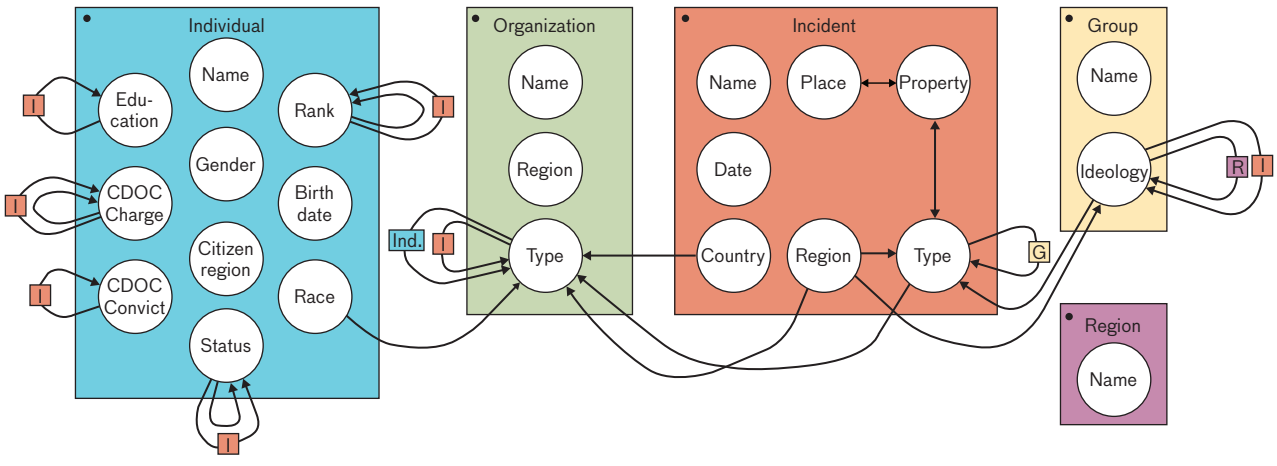

FIGURE 16. The relational dependency network learned from the ISVG data set is shown. Plates indicate node types (individual, organization, incident, group, and region). Circles in the plates indicate attributes of the nodes; e.g., rank of the individual. Edges between discs show the dependencies.

Future Directions

As the scale and ubiquity of unstructured, content-based data continue to increase so will the need for analytical tools to process and represent those data and to perform basic and reliable analytics, such as community detection, which support higher-level inference and prediction. Work done at Lincoln Laboratory in these areas is laying the groundwork for further research supporting these goals. Specifically, the Laboratory is continuing to investigate applying these tools and working with unstructured multimedia data, databases, graph-based analytics, and visualization. Further research and development will provide ways for users and stakeholders to better consume and summarize the massive amounts of information created every day in the era of Big Data. ■

References

1. N. Eagle and A. Pentland, "Reality Mining: Sensing Complex Social Systems," *Personal and Ubiquitous Computing,* vol. 10, no. 4, 2006, pp. 255–268.
2. C.K. Dagli and W.M. Campbell, "Individual and Group Dynamics in the Reality Mining Corpus," IEEE/ASE International Conference on Social Computing (SocialCom), 2012.
3. T.M.T. Do and D. Gatica-Perez, "Human Interaction Discovery in Smartphone Proximity Networks," *Personal and Ubiquitous Computing,* Dec. 2012, doi: 10.1007/s00779-011-0489-7.
4. D.O. Olguin, P.A. Gloor, and A.S. Pentland, "Capturing Individual and Group Behavior with Wearable Sensors," AAAI Spring Symposium on Human Behavior Modeling, 2009.
5. C. Weinstein, W.M. Campbell, B.W. Delaney, and G. O'Leary, "Modeling and Detection Techniques for Counter-Terror Social Network Analysis and Intent Recognition," *Proceedings of the IEEE Aerospace Conference,* 2009, pp. 1–16.
6. W.M. Campbell and Z. Karam, "Simple and Efficient Speaker Comparison Using Approximate KL Divergence," *Proceedings of Interspeech,* 2010, pp. 362–365.
7. D.O. Olguin, B.N. Waber, T. Kim, A. Mohan, K. Ara, and A.S. Pentland, "Sensible Organizations: Technology and Methodology for Automatically Measuring Organizational Behavior," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics,* vol. 39, no. 1, 2009, pp. 43–55.
8. J. Diesner and K. Carley, "Exploration of Communication Networks from the Enron Email Corpus," *Proceedings of the Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining,* 2005, pp. 3–14.
9. H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a Social Network or a News Media?" *Proceedings of the 19th International Conference on World Wide Web,* 2010, pp. 591–600.
10. *Institute for the Study of Violent Groups Codebook,* Sam Houston State University, Huntsville, Texas, 2007.
11. G. Doddington, A. Mitchell, M. Pryzbocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The Automatic Content Extraction (ACE) Program—Tasks, Data, and Evaluation," *Proceedings of the 2004 Conference on Language Resources and Evaluation,* 2004, pp. 837–840.
12. R.J. Brachman and H.J. Levesque, *Knowledge Representation and Reasoning,* San Francisco: Morgan Kaufmann, 2004.
13. S. Russell, P. Norvig, J. Canny, J. Malik, and D. Edwards, *Artificial Intelligence: A Modern Approach, vol. 2,* Englewood Cliffs, N.J.: Prentice Hall, 1995.
14. E. Codd, "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM,* vol. 13, no. 6, 1970, pp. 377–387.
15. J. Broekstra, A. Kampman, and F. Van Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema," *Proceedings of the First International Conference on the Semantic Web,* 2002, pp. 54–68.
16. A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark Graphs for Testing Community Detection Algorithms," *Physical Review E,* vol. 78, no. 4, 2008, pp. 046110-1–5.
17. M.E.J. Newman and M. Girvan, "Finding and Evaluating Community Structure in Networks," *Physical Review E,* vol. 69, no. 2, 2004, pp. 026113-1–15.
18. M.E.J. Newman, "Fast Algorithm for Detecting Community Structure in Networks," *Physical Review E,* vol. 69, 2004, pp. 066133-1–5.
19. A. Lancichinetti and S. Fortunato, "Community Detection Algorithms: A Comparative Analysis," *Physical Review E,* vol. 80, 2009, pp. 056117-1–12.
20. A. Clauset, M.E.J. Newman, and C. Moore, "Finding Community Structure in Very Large Networks," *Physical Review E,* vol. 70, no. 6, 2004, pp. 066111-1–6.
21. J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, 2000, pp. 888–905.
22. M. Rosvall and C.T. Bergstrom, "Maps of Random Walks on Complex Networks Reveal Community Structure," *Proceedings of the National Academy of Sciences,* vol. 105, no. 4, 2008, pp. 1118–1123.
23. K. Carley, "Dynamic Network Analysis," *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers,* R. Breiger, K. Corley, and P. Pattison, eds. Washington, D.C.: The National Academies Press, 2003, pp. 133–145.
24. M. Lahiri and T. Berger-Wolf, "Mining Periodic Behavior in Dynamic Social Networks," *Proceedings of the 8th IEEE International Conference on Data Mining,* 2008, pp. 373–382.
25. J. Sun, C. Faloutsos, S. Papadimitriou, and P. Yu, "Graphscope: Parameter-Free Mining of Large Time-Evolving Graphs," *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2007, pp. 687–696.
26. L. Akoglu and C. Faloutsos, "Event Detection in Time Series of Mobile Communication Graphs," 27th Army Science Conference, December 2010.
27. B. Bader, R. Harshman, and T. Kolda, "Temporal Analysis of Semantic Graphs Using ASALSAN," *Proceedings of the 7th IEEE International Conference on Data Mining,* 2007, pp. 33–42.

28. J. Sun, C. Tsourakakis, E. Hoke, C. Faloutsos, and T. Eliassi-Rad, "Two Heads Better Than One: Pattern Discovery in Time-Evolving Multi-aspect Data," *Data Mining and Knowledge Discovery,* vol. 17, no. 1, 2008, pp. 111–128.
29. J. Sun, D. Tao, and C. Faloutsos, "Beyond Streams and Graphs: Dynamic Tensor Analysis," *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2006, pp. 374–383.
30. K. Farrahi and D. Gatica-Perez, "What Did You Do Today?: Discovering Daily Routines from Large-Scale Mobile Data," *Proceedings of the 16th ACM International Conference on Multimedia,* 2008, pp. 849–852.
31. N. Eagle and A. Pentland, "Eigenbehaviors: Identifying Structure in Routine," *Behavioral Ecology and Sociobiology,* vol. 63, 2009, pp. 1057–1066.
32. N. Eagle, A. Pentland, and D. Lazer, "Inferring Friendship Network Structure by Using Mobile Phone Data," *Proceedings of the National Academy of Sciences,* vol. 106, no. 36, 2009, pp. 15274–15278.
33. J. Reades, F. Calabrese, A. Sevtsuk, and C. Ratti, "Cellular Census: Explorations in Urban Data Collection," *IEEE Pervasive Computing,* vol. 6, no. 3, 2007, pp. 30–38.
34. T. Kolda and J. Sun, "Scalable Tensor Decompositions for Multi-aspect Data Mining," *8th IEEE International Conference on Data Mining,* 2008, pp. 363–372.
35. MIT Academic Calendar 2004–2005, online, 2004. Available at http://web.mit.edu/registrar/ www/calendar0405.html.
36. MIT Fall 2004–2005 Class Schedule, online, 2004. Available at http://replay.waybackmachine.org/20041124034024/http:/web.mit.edu/registrar/www/ schedules/csbindex.shtml#15.
37. MIT 2004–2005 Course Catalog, online, 2004. Available at http://web.mit.edu/catalog/ archive/2004-2005/part 3.pdf.
38. B. Delaney, A. Fast, W. Campbell, C. Weinstein, and D. Jensen, "The Application of Statistical Relational Learning to a Database of Criminal and Terrorist Activity," *Proceedings of 2010 SIAM Conference on Data Mining*, 2010.
39. H. Blau, N. Immerman, and D. Jensen, "A Visual Language for Querying and Updating Graphs," University of Massachusetts–Amherst, Technical Report 2002-37, 2002.
40. M. McPherson, L. Smith-Lovin, and J. Cook, "Birds of a Feather: Homophily in Social Networks," *Annual Review of Sociology,* vol. 27, no. 1, 2001, pp. 415–444.
41. D. Jensen and J. Neville, "Linkage and Autocorrelation Cause Feature Selection Bias in Relational Learning," *Proceedings of the 19th International Conference on Machine Learning,* 2002, pp. 259–266.
42. D. Jensen, J. Neville, and M. Hay, "Avoiding Bias When Aggregating Relational Data with Degree Disparity," *Proceedings of the 20th International Conference on Machine Learning,* 2003, pp. 274–281.
43. L. Getoor and B. Taskar, eds., *Introduction to Statistical Relational Learning.* Cambridge, Mass.: The MIT Press, 2007.
44. D. Jensen, J. Neville, and B. Gallagher, "Why Collective Inference Improves Relational Classification," *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2004, pp. 593–598.
45. P. Sen, G.M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective Classification in Network Data," *AI Magazine,* vol. 29, no. 3, 2008, pp. 93.
46. J. Neville, D. Jensen, L. Friedland, and M. Hay, "Learning Relational Probability Trees," *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2003, pp. 625–630.
47. F. Provost and P. Domingos, "Tree Induction of Probability-Based Ranking," *Machine Learning,* vol. 52, no. 3, 2003, pp. 199–215.
48. A. Fast, L. Friedland, M. Maier, B. Taylor, D. Jensen, H.G. Goldberg, and J. Komoroske, "Relational Data Pre-processing Techniques for Improved Securities Fraud Detection," *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2007, pp. 941–949.
49. J. Neville, O. Simsek, D. Jensen, J. Komoroske, K. Palmer, and H. Goldberg, "Using Relational Knowledge Discovery to Prevent Securities Fraud," *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2005, pp. 449–458.
50. J. Neville and D. Jensen, "Relational Dependency Networks," *Journal of Machine Learning Research,* vol. 8, 2007, pp. 653–692.
51. D. Heckerman, D. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency Networks for Inference, Collaborative Filtering, and Data Visualization," *Journal of Machine Learning Research,* vol. 1, 2001, pp. 49–75.
52. G. Casella and E. I. George, "Explaining the Gibbs Sampler," *The American Statistician,* vol. 46, no. 3, 1992, pp. 167–174.

About the Authors

William M. Campbell is a senior staff member in the Human Language Technology Group. Prior to joining Lincoln Laboratory, he worked on speech processing and communication systems at Motorola. At Motorola, he worked on a variety of commercial and government projects involving speaker recognition (CipherVox, ETSI standards), voice coding (Tenor Pager), speech recognition (Land Warrior), spread-spectrum communication, and channel coding. Since joining Lincoln Laboratory in 2002, he has worked on speech processing, machine learning, and social network methods. His major contributions in speaker and language recognition have been widely cited and used in operations. In the area of social networks, he has made numerous contributions in graph analysis—simulation, machine learning, and construction of networks from multimedia content. He is the author of 100 peer-reviewed papers and has 14 patents. He received three bachelor's degrees—in computer science, electrical engineering, and mathematics—from the South Dakota School of Mines and Technology. He received master's and doctoral degrees from Cornell University in applied mathematics with a minor in electrical engineering.

Charlie K. Dagli has been a member of the research staff in the Human Language Technology Group at Lincoln Laboratory since January 2010. His primary research interests are in multimedia search, biometrics, and social network analysis. Prior to joining the Laboratory, he held positions at Hewlett-Packard Laboratories, Ricoh Innovations, and State Farm Corporate Research. He was the recipient of the Best Student Paper award at the 2006 ACM International Conference on Image and Video Retrieval and is a member of the IEEE. He received a bachelor's degree from Boston University in 2001, and master's and doctoral degrees from the University of Illinois, Urbana-Champaign, in 2003 and 2009, all in electrical and computer engineering.

Clifford J. Weinstein has served as leader of the Human Language Technology (HLT) Group and its predecessors continuously since 1979. He has made technical contributions to and led research programs in speech recognition, speech coding, machine translation, speech enhancement, social network analysis, packet speech communications, information assurance and cyber security, integrated voice and data communication networks, digital signal processing, and radar signal processing. His early work and landmark 1983 paper on packet speech helped lead to Lincoln Laboratory's first IEEE Milestone, awarded in 2011 for "First Real-Time Speech Communication on Packet Networks." He received the MIT Lincoln Laboratory Technical Excellence Award in 2013 for his nationally recognized leadership in the field of human language technology and his technical contributions to a broad spectrum of communications technologies, including digital signal processing, speech communications in packet networks, speech recognition and machine translation, and automated social network analysis. He was elected a Fellow of the IEEE in 1993 for leadership in speech recognition, packet speech, and integrated voice/data networks. He holds bachelor's, master's, and doctoral degrees in electrical engineering from MIT, and joined Lincoln Laboratory as an MIT graduate student research assistant in 1967.