Recommender Systems for the Department of Defense and Intelligence Community

Vijay N. Gadepally, Braden J. Hancock, Kara B. Greenfield, Joseph P. Campbell, William M. Campbell, and Albert I. Reuther

Recommender systems, which selectively filter information for users, can hasten analysts' responses to complex events such as cyber attacks. Lincoln Laboratory's research on recommender systems may bring the capabilities of these systems to analysts in both the Department of Defense and intelligence community.

In the past five years, the machine learning and artificial intelligence communities have done significant work in using algorithms to identify patterns within data. These patterns have then been applied to various problems, such as predicting individuals' future responses to actions and performing pattern-of-life analysis on persons of interest. Some of these algorithms have widespread application to Department of Defense (DoD) and intelligence community (IC) missions. One machine learning and artificial intelligence technique that has shown great promise to DoD and IC missions is the recommender system, summarized by Resnick and Varian [1], and its extensions described by Adomavicius and Tuzhilin [2]. A recommender system is one that uses active information-filtering techniques to exploit past user behavior to suggest information tailored to an end user's goals. In a recent working paper [3], the Office of the Director of National Intelligence's Technical Experts Group's Return on Investments team has identified recommender systems as a key "developing application" in their process map of "The Intelligence Cycle and Human Language Technology." The most common domain in which recommender systems have been used historically is commerce: users are customers and the objects recommended are products. Other feasible uses for recommender systems include recommending actions, e.g., suggesting a direct traffic route, and following interactions between users, e.g., proposing possible colleagues as the popular service LinkedIn does.

In the cyber arena, recommender systems can be used for generating prioritized lists for defense actions [4], for detecting insider threats [5], for monitoring network security [6], and for expediting other analyses [7].

Elements of Recommender Systems

Recommender systems rely on four important elements:

- *Information filtering*. Recommender systems do not singlehandedly convert data to knowledge; they are just one component of the information pipeline. Sensors collect data, data processing turns those bytes of data into useful pieces of information, and then recommender systems help to filter that information into the most relevant pieces from which a human can extract knowledge and take action. Note that filtering referred to here does not imply deletion of any information but rather prioritization.
- *User behavior*. The value of having computers learn from user behavior rather than apply prescribed rules or heuristics is that the users are never required to explicitly state what the rules are. The rules by which users make decisions are inferred from the way the users act. This utilization of user behavior rather than heuristics enables recommender systems to reflect nuances in individual human preferences that would otherwise be difficult to quantify. It also provides us with a simple test for classification of decision support systems: if a system makes recommendations that do not include considerations of past user behavior, then it is not a recommender system.
- *Suggest information*. Recommender systems operate under a "push" rather than a "pull" paradigm. An information-retrieval system, such as a search engine, is guided by a query submitted by the user—a pull for information. Recommender systems, on the other hand, utilize user behavior and context history to ascertain the needs of users and are therefore equipped to predict or prescribe, i.e., push, new information to the user.
- *End user goals*. The main distinction between recommender systems and the broader class of filtering and sorting techniques is the applicability of the output of a recommender system to the needs of a particular user or group of similar users.

Recommender systems consist of four primary components: users, objects, ratings, and a model. Users include anyone whose behavior is being recorded in some way to train the recommender system or anyone who is receiving recommendations. Objects refer to products, documents, courses of action, or other recommendations. Ratings are some quantifiable measure of the utility of a given user-object pair and may come from explicit feedback (e.g., thumbs-up votes, assessments on a five-star rating scale, or text reviews) or implicit feedback (e.g., number of clicked links, number of downloaded files, or time spent on a page). The model is used to process known ratings and make recommendations based on the predicted ratings for unrated user-object pairs. The functional architecture depicted in Figure 1 shows these four elements, with additional detail shown for the four stages in the workflow of a generic model.

Recommender systems are able to make relevant suggestions in a given situation by observing how users act (i.e., recording ratings assigned to particular objects in a specific context). How users act is, in turn, affected by the goals or policy that the user follows, the user's intuition about what objects will satisfy those goals, and domain-specific knowledge that the user may have. This information is generally not formalized or conveyed to the recommender system. However, as the recommender learns from the user behavior that is affected by these influences, its recommendations will begin to reflect these influences.

User behavior is recorded in the data collection stage of a recommender system. In addition to ratings, information about traits of users, such as range of ratings or scores, objects, or context may be recorded. Context denotes situation parameters that can be known by the system and may have an impact on the selection and ranking of recommendation results.

Once these data have been collected, they are used to update the model of user preferences. First, significant features such as important aspects of a dataset (e.g., in a cyber network log, one feature may be time of logged event) must be extracted. Explicit user ratings may be entered directly, whereas implicit ratings may require some processing or inference to relate user behavior to a quantifiable rating to be stored. To reduce noise and lower computational complexity, some form of dimensionality reduction (i.e., a mechanism to reduce the number of variables being considered to the most critical variables) is often performed at this stage.

Once the model is updated, the next task is to estimate the ratings that the current user would give to the objects

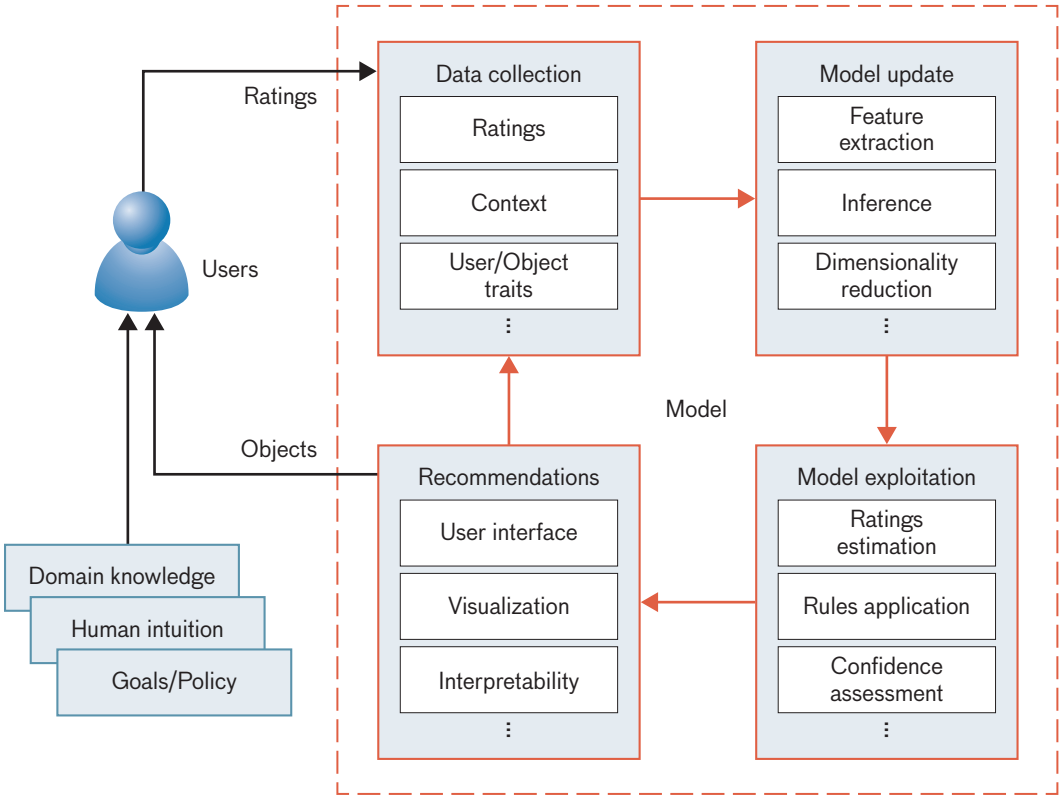

FIGURE 1. A recommender system consists of users, objects, and ratings that interact with each other through a model developed by the recommender system. A recommender system takes information collected from domain knowledge, human intuition and goals, or policy and combines that information with user ratings. The recommender model is derived from data collections, model updates, model exploitation, and recommendations from other users or previous actions.

with which he or she has not yet interacted. Collaborative filtering and content-based, knowledge-based, or hybrid techniques (described in the next section) are used to generate these rating estimates. At this stage, additional recommendation rules, such as favoring recommendations that support specific objectives, may be applied. For example, a commercial recommender may be designed to favor products with large profit margins.

Then, using the estimation just performed, the recommender returns results to the user in a desired form (e.g., a top result, top n list of results, or all results above a given threshold). The subsequent actions of the user are recorded, and the cycle repeats.

How a Recommender System Works

To illustrate how a recommender system works, let us look at a very simple recommender system that recommends online articles or documents for an analyst to examine. In this example (modified from Jannach et al. [8]), we are applying a collaborative-filtering recommender system in which analysts are searching through a corpus of online documents for information about potential exploits or cyber attacks. In this scenario, because the number of documents is greater than the number of analysts, the analysts rely on a recommender system to prioritize important documents. For ease of illustration, we will consider only five analysts and six online documents although a real-world system could easily consist of many millions of analysts and documents. Assume that whenever an analyst reads a document, that document is given a rating on a scale of 1 to 5 (1 = not useful at all, 5 = very useful). This rating may come from both explicit analyst input and implicit input. Shown in Table 1 are the analyst ratings for the documents.

In this illustration, we want to predict what the rating of Analyst 1 would be for Doc 5 and Doc 6. The document

Table 1. Analysts' Document Ratings

	DOC 1	DOC 2	DOC 3	DOC 4	DOC 5	DOC 6
Analyst 1	5	3	4	4	?	?
Analyst 2	3	1	2	3	3	1
Analyst 3	3	3	1	5	4	5
Analyst 4	4	3	4	3	4	2
Analyst 5	1	5	5	2	1	3

Table 2. Similarity Scores Between Analysts via Adjusted Cosine Similarity

	ANALYST 1	ANALYST 2	ANALYST 3	ANALYST 4	ANALYST 5
Analyst 1	–	0.85	0.00	0.71	−0.79
Analyst 2		–	0.43	0.30	−0.89
Analyst 3			–	−0.71	−0.59
Analyst 4				–	−0.14
Analyst 5					–

with the higher rating can then be recommended to her to read next. The predicted document ratings for Analyst 1 will be based on the ratings given to those documents by analysts who have expressed similar ratings to hers in the past on other documents that they all have rated. This prediction will require some metric for measuring the similarity between users. Common metrics, many of which are described in Herlocker et al. [9], include cosine similarity, Pearson's correlation coefficient, Spearman's rank correlation coefficient, or the mean squared error difference. We will use a variant of cosine similarity called adjusted cosine similarity.

If we represent the ratings of a particular analyst by a vector, the cosine similarity of two vectors (ratings of two different analysts) is equal to their dot product, divided by the product of their magnitudes:

$$sim(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}|*|\vec{b}|}$$

The cosine similarity of Analyst 1 and Analyst 2 for the first four documents can then be calculated as

$$sim(A1,A2) = \frac{(5*3)+(3*1)+(4*2)+(4*3)}{\sqrt{5^2+3^2+4^2+4^2}*\sqrt{3^2+1^2+2^2+3^2}} = 0.975$$

This calculation, however, does not take into account that analysts may generally give ratings in different ranges. For example, a particular analyst may tend to give ratings only in the range of 3 to 5 or may click more links than most analysts click on every page he visits (leading to higher implicit ratings). These types of factors may be accounted for by subtracting from each rating the average rating given by that user. Using this adjusted cosine similarity formula, we obtain the similarity scores shown in Table 2 in which a higher score indicates greater similarity.

Because we are basing Analyst 1's unknown ratings on the ratings of those who have similar rating histories, we may choose to use the ratings of only the k closest

neighbors or of those who have similarity scores above a certain threshold. In this case, we will use the ratings of only those who have Analyst 1 similarity scores greater than or equal to 0.5 (Analyst 2 and Analyst 4). The predicted rating of document d for user a thus becomes

$$r_{a,d} = \overline{r_a} + \frac{\sum_{b \in k} sim(a,b) * \left(r_{b,d} - \overline{r_b}\right)}{\sum_{b \in k} sim(a,b)}$$

From this equation, we obtain the following values for $r_{A1,Doc5}$ and $r_{A1,Doc6}$:

$$r_{A1,\,Doc5} = 4 + \frac{0.85 * (3 - 2.17) + 0.71 * (4 - 3.33)}{0.85 + 0.71} = 4.75$$

$$r_{A1,\,Doc6} = 4 + \frac{0.85 * (1 - 2.17) + 0.71 * (2 - 3.33)}{0.85 + .071} = 2.97$$

Comparing the magnitudes of these predicted ratings reveals that Doc 5 should be recommended to Analyst 1 over Doc 6.

Example Recommender System Using Topic Modeling

A common form of a recommender system can use a mechanism of topic modeling to recommend objects such as new webpages, articles, or movies. The idea behind such a recommender system is that if a user is interested in a particular topic, she will be interested in other objects with the same topics. Similar to using techniques employed in a content-based recommender system, one may model a corpus of documents to find important topics. Once a topic is highlighted, a user is recommended other documents containing similar topics or terms. In this section, we will describe a simple but powerful way to perform topic modeling on very large datasets.

Non-negative matrix factorization (NMF), described by Lee and Seung [10], is a technique used to factorize a given matrix into two matrices, both of which only consist of non-negative elements. Multiplying these two matrices produces an approximation of the original matrix. Consider a matrix $A_{m \times n}$ to be factored into matrices $W_{m \times k}$ and $H_{k \times n}$, where m corresponds to the number of rows of A, n corresponds to the number of columns in A, and k corresponds to the number of topics. By definition,

$$A = W * H$$

In the above factorization, the columns of W can be considered a basis for the matrix A with the rows of H being the associated weights needed to reconstruct A. A common method to solve this factorization problem is through the alternating least-squares (ALS) algorithm as described in Gadepally et al. [11]. However, one of the challenges in working with very large datasets is the inability to store intermediate products produced by the ALS algorithm. Very often, intermediate matrices created in each iteration of the ALS algorithm can be many times larger than the original dataset or available computational resources.

We have recently developed a new tool to perform NMF on large, sparse datasets. We refer to this tool as the projected ALS. In addition to removing non-negative elements in each iteration of the ALS algorithm, we can also enforce a particular sparsity level. This method has been shown to perform qualitatively as well as the original dense ALS algorithm. However, with this extra projection step that enforces sparsity, we are able to achieve much better computational performance as shown in Figure 2. By computing the matrix factorization through the projected ALS algorithm, we can determine a set of topics from a data corpus. We applied the projected ALS algorithm to a corpus of data collected from the popular social media site Twitter. We then found five topics from this dataset (Table 3). If a user highlights a certain tweet, a recommender system can then find other tweets that have keywords within the same topics of the selected tweet. For example, if the user highlights a tweet with the word "love," the recommender system can suggest tweets with the hashtag "#PerksOfDatingMe" because these tweets are related via topic 2. This technique can be easily extended to the identification of malicious conversations about cyber attacks or other cyber events that often occur on social media sites such as Twitter; once the system discovers suspicious conversations, it can alert analysts to take a closer look at the suspect tweets.

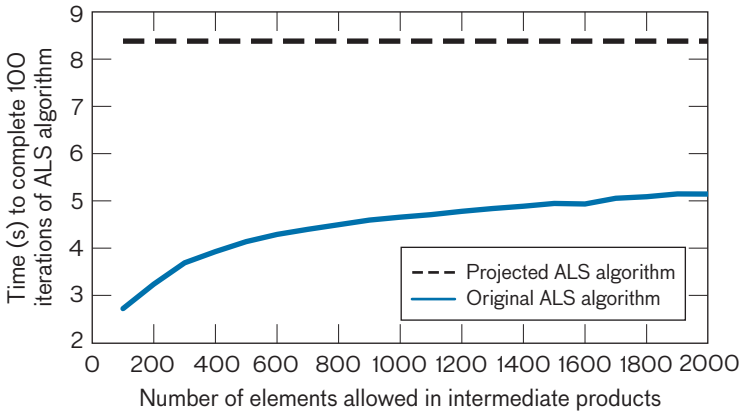

FIGURE 2. The time in seconds taken for 100 iterations of the alternating least-squares (ALS) algorithm. The dashed black line corresponds to the time taken for the original ALS algorithm that yields dense intermediate products. The solid blue line corresponds to the time taken for 100 iterations of the projected ALS algorithm that enforces sparsity within each iteration. The large reduction in time is due to the computational efficiency of the projected ALS algorithm.

Table 3. Topics in Twitter Posts Determined by Alternating Least-Squares Algorithm

TOPIC 1 (TWEETS WITH TURKISH WORDS)	TOPIC 2 (TWEETS RELATED TO DATING)	TOPIC 3 (TWEETS RELATED TO ACOUSTIC GUITAR COMPETITION IN ATLANTA, GEORGIA)	TOPIC 4 (TWEETS WITH SPANISH WORDS)	TOPIC 5 (TWEETS WITH ENGLISH WORDS)
word|:)	word|#PerksOfDatingMe	word|#5sosacousticATL	word|con	word|I'll
word|@	word|@	word|#5sosfam	word|creo	word|I've
word|Airport	word|My	word|#5sosgettoatlanta	word|cuando	word|If
word|Hastanesi	word|go	word|@5SOS	word|da	word|Just
Word|International	word|love	word|acoustic	word|del	word|Lol
word|Kadiköy	word|out	word|atlanta?	word|dormir	word|My

Recommender Systems and the Cyber Domain

Recommender systems have the potential to greatly reduce the response time to cyber threats. In the cyber domain, it is very easy for analysts to be inundated with information. For example, the Target Corporation's security breach was reported by the company's security software but was ignored along with many false positive alerts [12]. In such enterprise environments, recommender systems can be valuable tools to filter and prioritize information that may be of interest to an analyst. Consider the common case of an information technology (IT) security team defending an organization against evolving cyber threats. As reported by the 2015 Global Information Security Workforce Study [13], 62% of organizations claim that their information security teams are too small. These resource-constrained teams are also often responsible for paying attention to 100s of websites and blogs to look for information about publicly reported exploits. These teams may then have to turn to the National Vulnerability Database [14] to understand the impact of exploits to their organization. Finally, these teams may develop patches that are eventually deployed across the organization with varying levels of impact to the end users.

As of 2015, approximately 25 new vulnerabilities are reported per working day (as calculated by using

the number of Common Vulnerabilities and Exposures listed in the National Vulnerability Database [14] for the year 2015). In developing an appropriate response, the security team must weigh dozens of factors, such as the time since the vulnerability's discovery, severity of the exploit, existence of a patch, difficulty of deploying the patch, and impact of the patch on users. Recommender systems can provide a mechanism to greatly simplify this response process. A recommender system can automatically track 100s of sites, learn from past user behavior about important cyber security news items, and recommend them to the IT security team. Recommender systems can use prior information about the vulnerability's severity and impact to the user community to suggest a course of action for patching the vulnerability. For example, a recommender system may propose postponing the deployment of a minor vulnerability's patch that would cause a major impact for the user community, or the system may recommend immediate deployment of a major vulnerability's patch that would have minor user impact. Furthermore, recommender systems can be used to track anomalies across the network (such as unpatched systems or systems exhibiting behavior very different from that of others on a network) to allow the limited resources of the IT security team to quickly address potentially important problems rather than being inundated with regular traffic.

Specific Concerns of the Department of Defense and Intelligence Community

While recommender systems have reached maturity in the commercial world, there are many challenges in directly applying these systems to DoD and IC problems. Commercial and government entities both have the need to collect, store, and process a large amount of high-dimensional data. However, government applications have certain traits that make utilizing traditional methods to produce actionable intelligence more difficult. Some of these differences are shown in Table 4.

The first difference concerns the lack of ground truth and the difficulty in quantifying success for DoD applications. In industry, success tends to be measured by a concrete action, such as a sale of a product or a click on a webpage. In DoD and IC applications, however, the desired measure of effectiveness is whether or not an action will lead to a greater probability of mission success. Because this metric is speculative, it is much more difficult to measure than the commercial standard of profitability.

Compared to industry applications, government applications typically carry much more extreme consequences for false automatic calculations that lead to suboptimal decisions. Once again, the magnitudes of these consequences are harder to quantify. Dollars provide an obvious surrogate for risk in a commercial setting, but there is no clear, established metric for measuring operational readiness. The lack of such a metric makes it difficult to determine whether government organizations should use a particular piece of technology in support of their mission.

Similar to commercial cyber security applications, government applications exist in a space where the adversary is continually evolving. Yet, the current architectural and political landscape found in most government organizations necessitates that analytics are developed, deployed, and re-engineered over a much longer time scale than industrial applications typically employ. Thus, government organizations experience fewer opportunities to make incremental improvements to the underlying analytics.

Differences in the skill levels of users affect the design and value of recommender systems. Users of recommender systems in the DoD and IC will likely be experts in their fields who engage with these systems daily. Such familiarity with the system may allow for more capable and complex functionality to be utilized. Perhaps, more importantly, the inclusion of experts in this human-in-the-loop process may lead to a different balance of autonomy. Recommender systems may need to be capable of making the reasons behind their recommendations transparent in order to gain the confidence of experts who are making high-stakes decisions. For example, a system may provide the end user with a confidence measure (such as probability) associated with each recommendation.

Another significant difference between big data applications used by commercial and government groups is that a commercial entity generally controls its data sources and approaches the data with specific goals and questions while government groups usually do not. For instance, Google may create a new feature on Google+ to obtain a different type of information from its user base to better its advertising services. However, government agencies, which usually do not have collection authority over the data they use, have limited control over designing data collection

Table 4. Comparison of Commercial Applications to DoD Applications

COMMERCIAL APPLICATIONS	DOD APPLICATIONS
High dimensionality of data	High dimensionality of data
Large volume of data	Large volume of data
Known truth; easier to quantify success	Unknown truth; difficult to quantify success
Mild consequences of decisions	Serious consequences of decisions
Past is representative of future	Past does not represent future
Continual development and improvement	Deployment; long durations between improvements
Average or untrained users	Expert or trained users

paradigms; even those agencies with collection authority are often bound by regulations that restrict their ability to deploy sensors that can collect the specific data they desire.

Finally, commercial applications are often designed to learn from millions to billions of users whereas DoD and IC applications may only have 100s to 1000s of users whose behaviors can be used to model recommender systems. Also, very often, commercial entities employ user agreements to determine data collection and usage whereas government organizations may not be able to readily access data without the help of law enforcement or legal statutes.

Recommender Systems Applied to Lincoln Laboratory Programs

Lincoln Laboratory has a rich history in developing decision support systems. Over the past five years, these systems have been incorporating recommender system concepts and technologies. In this section, we summarize past, current, and future Laboratory programs that incorporate recommender systems. The work conducted in these programs can inform research into systems that improve cyber security.

Dynamic Customization of Content Filtering

The objective of Dynamic Customization of Content Filtering (DCCF) is to allow an analyst to perform on-the-fly customization of content filtering (for example, open-source social media data mining) on the basis of simple relevance feedback acquired during the inspection of filtered content (Figure 3).

First, the analyst sets the parameters of an initial data-stream filter (e.g., keywords, geographical area, time interval) to mine for content of interest. Typically, as when keyword filters are used on social media data, this approach will lead to a mixture of relevant content embedded within various types of irrelevant content. While reviewing the content, the analyst provides simple binary feedback (indicating relevance or irrelevance) as desired and submits this feedback to the system. The DCCF model uses this feedback to create a secondary filter to remove irrelevant data that passes through the first filter. The creation of this secondary filter is based on a broad set of text- and image-derived feature spaces (i.e., characteristics of a general dataset; a dataset of a network's cyber attacks may include feature spaces such as date, time, type) coupled with aggressive feature space downselection and classifier training so that the model is suited to potentially diverse content-filtering needs. The DCCF model is generated on the fly (during analyst use) every time new feedback is submitted, thus improving content filtering as the user increasingly interacts with DCCF. The DCCF tool may be considered a recommender system because it pushes out filtered content that is based on earlier user-specific feedback.

Delve

The goal of Delve is to develop an approach for recommending documents to analysts who are answering broad, complex questions. This task is particularly suited for recommender systems because analysts are often uncertain as to what relevant information may be available to them

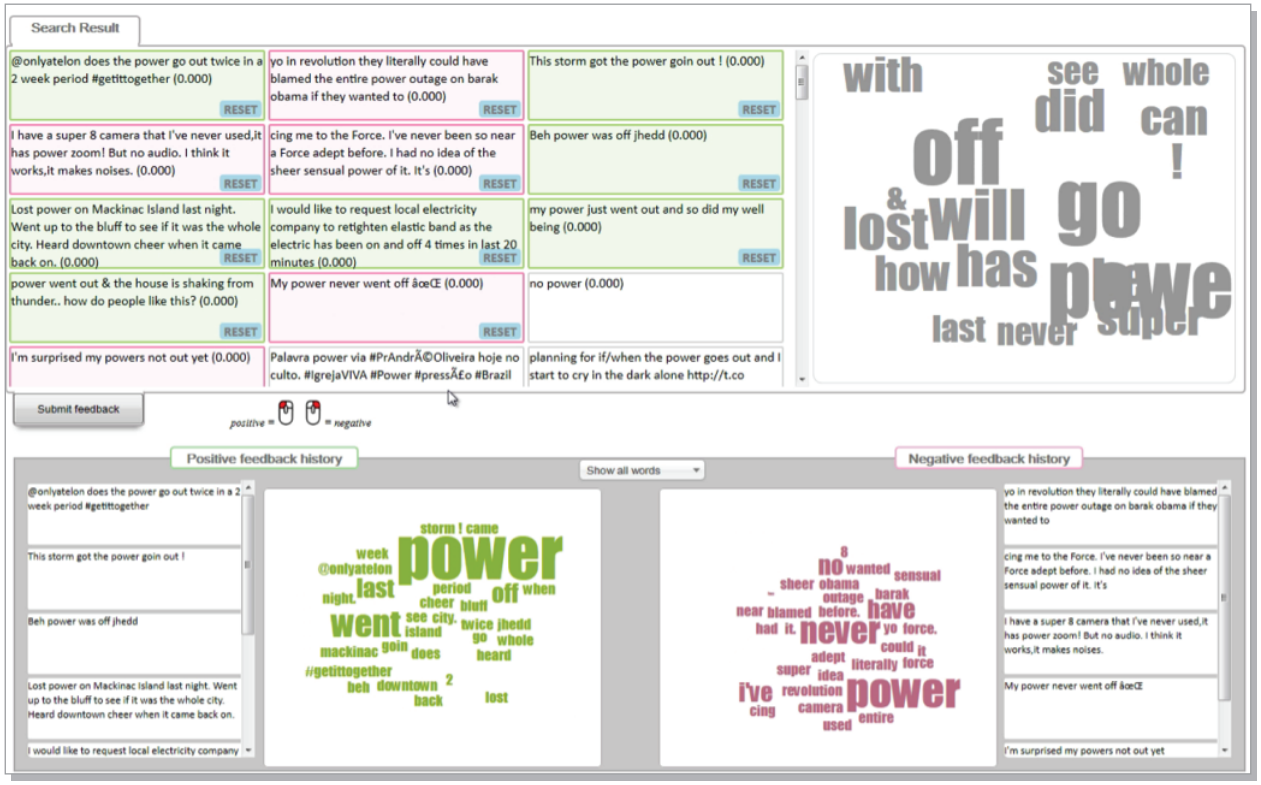

FIGURE 3. The Dynamic Customization of Content Filtering (DCCF) uses analyst feedback to perform on-the-fly customization of content filtering. A user first sets parameters such as keywords, geographical area, or time interval. The DCCF model will automatically filter content on the basis of these keywords to expedite retrieval of useful content. The word cloud on the top right corresponds to all retrieved results from the keyword filter. The bottom word clouds correspond to results from the secondary filters.

and therefore are ill-equipped to find all the information that they need via precise queries only. The Delve system employs a hybrid recommender that calculates both individual document characteristics (e.g., word count, number of entities) and collective browsing behavior (e.g., identification of articles that tend to co-occur or follow others in a browsing path). Using these calculations along with dimensionality-reduction techniques, Delve significantly outperforms baseline approaches, such as using only webpage attributes or term frequency–inverse document frequency (TF-IDF), for recommending additional documents of interest, given an initial document selected by the analyst.

Global Pattern Search at Scale

The Global Pattern Search at Scale (GPSS) is a scalable visual analytics platform to support the analysis of unstructured geospatial intelligence. With GPSS, analysts can interactively explore the document corpus at multiple geospatial resolutions, identifying patterns that cut across various data dimensions, and can uncover key events in both space and time. The tool includes an interactive visualization featuring a map overlaid with document clusters and events, search and filtering options, a timeline, or a word cloud (Figure 4). As an information filtering tool, GPSS provides detail on demand. However, in its current form, GPSS does not "push" new intelligence information to the analyst. In the coming year, the GPSS team plans to augment their tool with a recommender system that will profile user activity and suggest new documents of interest after periods of inactivity, similar to the way that advanced news websites such as Google News will suggest articles related to topics, locations, or stories in which a specific reader has expressed past interest.

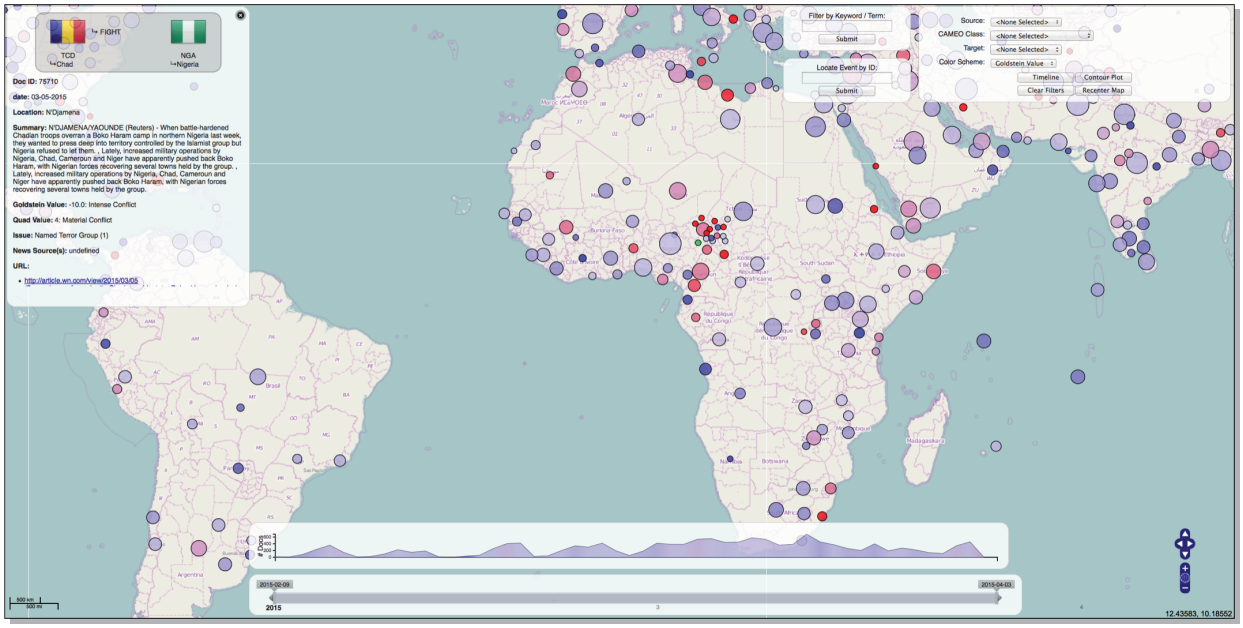

FIGURE 4. The Global Pattern Search at Scale (GPSS) platform enables interactive exploration of intelligence information by topic, time, and location. The GPSS system provides users with quick geospatial visualization about documents. In the display above, a user can search for terms, and the circles of different sizes and shades indicate, respectively, the prevalence of articles in a particular geospatial location (color spectrum runs from red for a region of conflict to lavender for a region of cooperation) at a particular time.

Covert or Anomalous Network Discovery and Detection

Networks are often used to describe relations or interactions between individuals, systems, or other entities via graphical models. The Covert or Anomalous Network Discovery and Detection (CANDiD) program aims to develop the mathematical understanding for constructing operationally relevant networks, detecting important subgraphs of these networks, and inferring and influencing the properties of select vertices in the network. Networks of interest often arise from large collections of complementary, redundant, and potentially noisy relational data sources, introducing challenges both in terms of algorithmic scalability and algorithmic accuracy. Through the CANDiD program, we are currently looking into applying a recommender system perspective to the problem of filtering and personalizing the multisource, noisy data used to construct and estimate the network of interest.

Adaptive, Reinforced, Interactive Visual Analytics

The goal of the Adaptive, Reinforced, Interactive Visual Analytics (ARIVA) program is to identify important information that aligns with analyst-provided feedback to better facilitate algorithmic-aided exploration of complex data for evolving open-ended missions, such as deterring cyber threats. User-provided feedback, in the form of similarity and dissimilarity assessments between pairs of data points, is utilized to perform data preprocessing via feature selection and transformation. When feedback-aligned data embeddings are accurately identified, common exploration analytics, such as data clustering, nearest-neighbor classification, and information-retrieval techniques, show algorithmic performance improvement and produce results that are grounded in an analyst's preferences, understanding of mission goals, and expertise in a given domain. The improvement of retrieval algorithms in feedback-aligned data spaces suggests that recommender systems can augment tools like ARIVA by utilizing the similarity or proximity of data points in the learned data embedding. The use of explicit pairwise similarity and dissimilarity constraints allows this application to avoid problems commonly found in recommendation engines whose limited feedback often leads to a reduction in recommender system performance.

Structured Knowledge Space

Structured Knowledge Space (SKS) is an end-to-end software system that combines information extraction, information retrieval, and natural language processing to intelligently explore a corpus of unstructured documents, such as intelligence reports of cyber threats. The SKS suite of tools extracts entities and creates structured metadata for each document to improve its searchability (Figure 5). With this metadata, analysts can find all documents that refer to a single organization or person (even when that entity has several aliases or variations), that contain a geospatial reference within a certain distance of a location, or that reference a time within a specified date range. Currently, SKS operates under a "pull" rather than a "push" paradigm (i.e., the user searches and browses rather than the system making recommendations). However, there are multiple ways in which recommender system concepts can be utilized to further enhance SKS. One enhancement would be to recommend new articles on the basis of past searches performed (e.g., "This article was recommended to you because of your interest in phishing attacks on enterprise networks."). Another enhancement would be to guide novice analysts' searches by using the search paths that more experienced analysts have taken.

Cyber Human Language Technology Analysis, Reasoning, and Inference for Online Threats

Through the Cyber Human Language Technology (HLT) Analysis, Reasoning, and Inference for Online Threats (CHARIOT) program, we are developing an interactive filtration system to automatically identify documents that are relevant to analysts' current investigations (Figure 6). With CHARIOT, analysts are presented with online discussions concerning cyber attack methods, defense strategies, and tools' effectiveness through the automated examination and classification of forum threads. CHARIOT leverages techniques such as topic classification, entity recognition, and sentiment analysis (i.e., opinion mining) to separate malicious cyber discussions from irrelevant discussions. The "Finding Malicious Cyber Discussions in Social Media" article in this issue discusses the CHARIOT program in further detail.

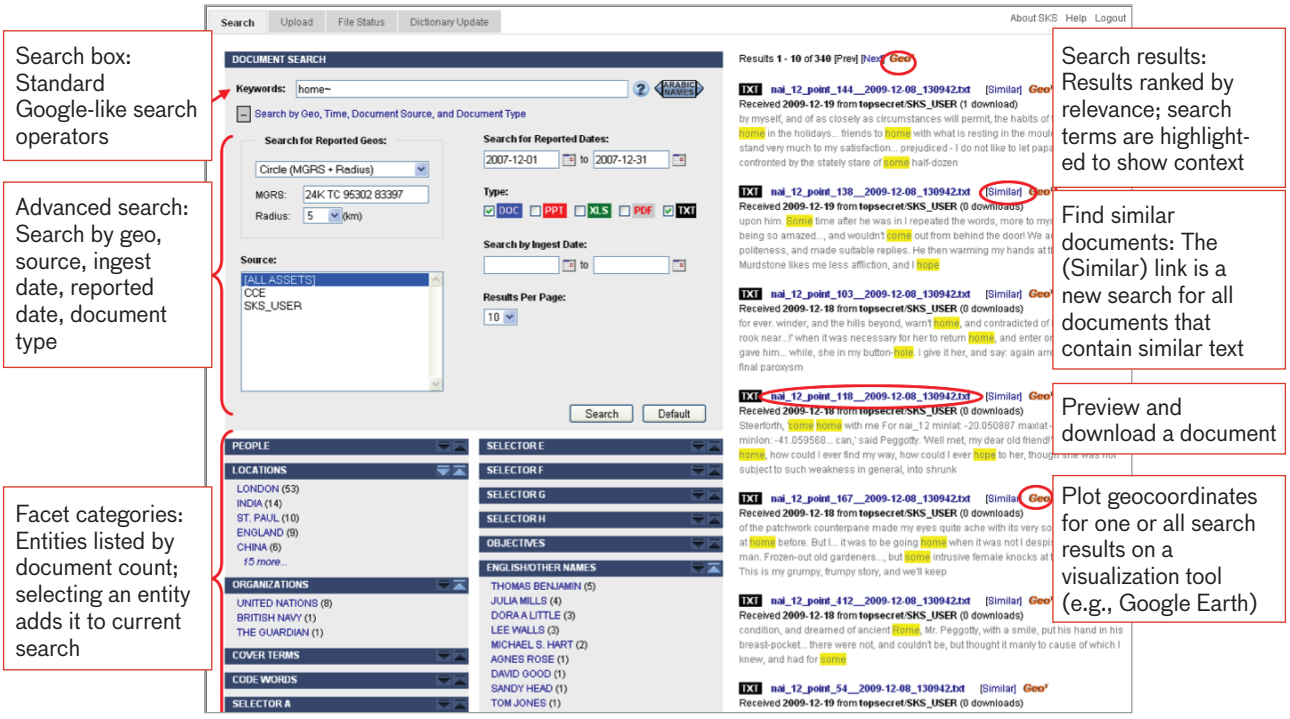

FIGURE 5. The Structured Knowledge Space search page provides diverse, useful information for the exploration of a corpus of unstructured documents.

Visualization, Summarization, and Recommendation for Multimedia

The goal of the Visualization, Summarization, and Recommendation (VISR) for Multimedia program is to develop tools to allow analysts to effectively explore large multimedia data sources. The program builds on previous Lincoln Laboratory work in text analytics by expanding to multimedia data, especially audio and video, with the addition of a recommendation component (Figure 7). This recommender system utilizes a user's ongoing work to identify other information of interest. For example, it may suggest videos of likely interest to an analyst on the basis of his or her current and past searches.

XDATA

The Defense Advanced Research Projects Agency's (DARPA) XDATA program aims to meet the challenges of big data analytics and visualization by developing computational techniques and software tools for processing and analyzing large, noisy, and incomplete data. For scalable analytics, this work includes research into distributed databases, statistical sampling methods, and new algorithmic advances to lower the computational complexity of pattern matching. For information visualization, this effort is focusing on the development of web-based human-computer interaction tools that factor computation between the client and the server and that are built from an open code base to enable rapid customization of tools to different missions. The XDATA program is investigating software that can efficiently fuse, analyze, and disseminate the massive volume of data these tools produce.

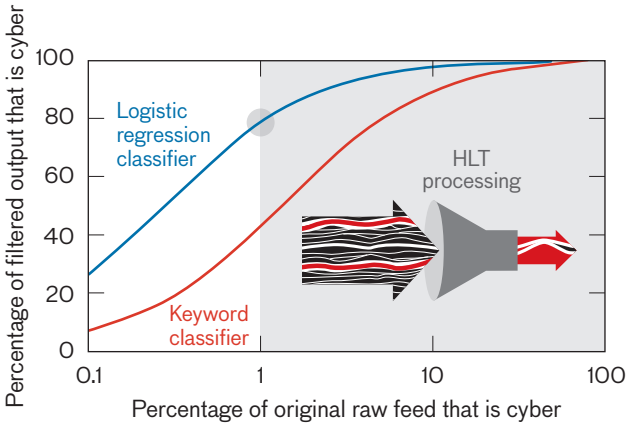

FIGURE 6. The goal of CHARIOT is to filter social media discussions to find cyber content. This figure contrasts a simple keyword classifier that detects cyber discussions via keywords or phrases with the CHARIOT-implemented logistic regression classifier. For an incoming data stream containing 1% cyber content, the CHARIOT logistic regression classifier can output a data stream containing 78% cyber content (compared to 43% for the keyword classifier).

Lincoln Laboratory's approach for the DARPA XDATA program is to provide key enabling technologies—including those for natural language processing, topic clustering, and text language identification—to extract information from structured, semistructured, and unstructured text, speech, image, and video data. This information is then used by the Laboratory and our partners for upstream (later in the development pipeline) analytics and visualization. The Laboratory has developed several analytics and user-interface technologies for graph query by example, entity

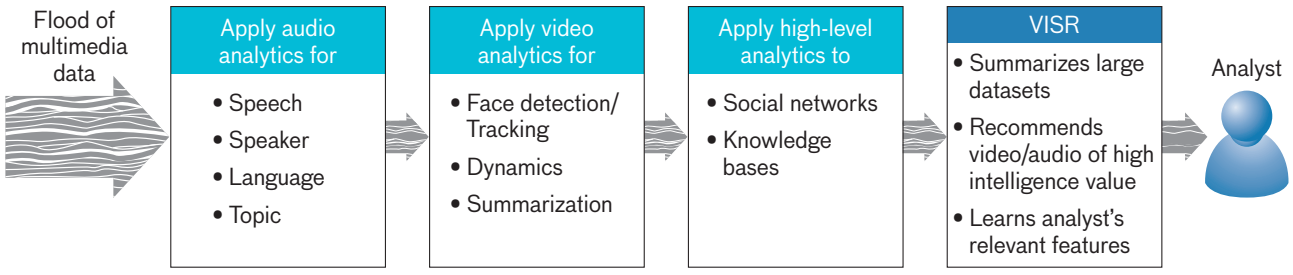

FIGURE 7. The Visualization, Summarization, and Recommendation (VISR) for Multimedia system takes a collection of multimedia audio and video data as an input. Notionally, three analytic processes at the individual item level process the data. First, standard speech processing is applied—speech, speaker, language, and topic recognition. Second, video analytics are applied to find faces and summarize object content. Third, high-level analytics are applied to find links between individual items. These links are user selectable. Examples of links include common faces, similar object content, or similar audio content. The VISR interface integrates all this functionality to display data to an analyst in a structured form for navigation and triage.

disambiguation, community detection, and correlation of similar network graph portions. Many of these enabling component technologies for recommender systems have been made publicly available via DARPA's Open Catalog of DARPA-sponsored software and peer-reviewed publications (http://opencatalog.darpa.mil/).

Future Work

Before the DoD and IC can employ recommender systems in operational settings, many technical and sociotechnical challenges must be overcome. We have identified six specific challenges that future work at Lincoln Laboratory and within the DoD and IC could productively address:

- *Establishing user trust.* The potentially serious consequences of decisions made via DoD applications necessitate a level of user trust in the recommender system. One way to increase trust is to enhance the interpretability or transparency of results, using algorithms that enable explanations to be given for why a particular object has been recommended, as discussed in O'Donovan and Smyth [15]. Another approach to fostering trust in a recommender system could be to verify the reliability of the source of data used in the development of the system's model by tracking the data's provenance, i.e., its origins and route of transfer.
- *Preserving privacy and security.* While there is certainly a need to make the recommendations of a recommender system transparent, there is simultaneously a potentially conflicting need to ensure the privacy of users, as described in Avesani et al. [16] and Brekovsky et al. [17]. A system that relies on tracking user history—sometimes in great detail (e.g., purchase history, browsing history, eye tracking)—has the potential to be misused by users to learn nonpublic details about other users if security precautions are not taken. Similarly, the security of the system may be at risk if individual users are able to reverse engineer the system to learn, for example, that submitting a certain number of specific inputs can ensure that another user will see a given output. Cryptographic techniques, such as those described by Gadepally et al. [18] and Shen et al. [19], may prove a useful means for building into recommender systems guarantees that prevent information from being discoverable by unauthorized users.
- *Adapting to user environment.* The types of users and usage contexts of recommender systems can reasonably be expected to vary significantly between commercial and defense applications. Whereas commercial systems tend to be used in environments that demand little user concentration, have few time constraints, and assume minimal user experience with the system, defense systems have the potential to be used in environments that require high concentration from users, adhere to strict deadlines, and employ operators who have been trained to engage with the system on an intricate level. Research into how existing mechanisms may be modified to address DoD and IC constraints or to exploit the capabilities of their personnel and systems seems prudent.
- *Developing multilevel metrics.* Of the many ways to assess the value of a recommender system, the majority of these assessments pertain to the perceived quality of recommendations. Some of these metrics are described by Gunawardana and Shani [20]. In addition to these recommendation-level metrics, however, there is also a need for system-level and user-level metrics. System-level metrics may reflect the measured time savings of a decision process or a change in the percentage of documents read that are considered relevant. User-level metrics consider the users' experiences with the system—how are concentration, decision fatigue, or confidence in decisions affected when users interact with the system? This area may overlap to some extent with the requirement of establishing user trust.
- *Promoting system extensibility.* While the core algorithms of recommender systems are often made public via publications and presentations, deployment and maintenance details are rarely discussed. From an institutional standpoint, it is important for the DoD and IC and their partners to understand how transferable the developed technology in this area will be from one domain or mission to another. It would be valuable to understand which technologies require domain-specific tuning and which ones can be rapidly deployed in new scenarios with little modification. Determining which pieces may be modularized for reapplication or redeployment could lead to improved cost estimates over the lifecycle of the developed technology. The development of a standardized recommender system application program interface could

lead to users' ability to immediately and easily interact with data in multiple forms on a variety of databases.

- *Developing partners in academia and industry.* The future research areas we have described focus on the specific needs of the DoD and IC. However, work in recommender systems encompasses a number of fields, including machine learning, big data analytics, and user experience, and many individuals in academia and industry are also conducting research in these fields. These researchers could partner with us to provide innovative ways to advance the role of recommender systems in various domains.

Recommender systems could have a significant impact in defense and intelligence applications. With the ability to learn from user behavior and push suggestions to users, they have the potential in mission scenarios to shift computational support from being reactive to being predictive. Recommender system technology has been advanced substantially in recent years by commercial entities, but some future work will be required to adapt these technologies for use in the defense domain, where requirements and objectives differ from those of commercial applications.

Acknowledgments

We would like to thank the Lincoln Laboratory staff members who contributed to this work: Paul Breimyer, Paul Burke, Rajmonda Caceres, R. Jordan Crouser, Matthew Daggett, Jack Fleischman, David Weller-Fahy, Vitaliy Gleyzer, Robert Hall, Andrew Heier, Stephen Kelley, David Martinez, Benjamin Miller, Paul Monticciolo, Kenneth Senne, Danelle Shah, Mischa Shattuck, Olga Simek, Steven Smith, William Streilein, Jason Thornton, Michael Winterrose, and Michael Yee. We would also like to thank research librarian Robert Hall for his help with the literature review and Marc Bernstein for supporting this work. A special thanks to Ariana Tantillo and Dorothy Ryan for their help in editing the article. ■

References

1. P. Resnick and H.R. Varian, "Recommender Systems," *Communications of the ACM*, vol. 40, no. 3, 1997, pp. 56–58.
2. G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, 2005, pp. 734–749.
3. S. Bailey, "The Intelligence Cycle and Human Language Technology," HLT Return on Investment Working Group, internal document, July 2015.
4. K.B. Lyons, "A Recommender System in the Cyber Defense Domain," master's thesis no. AFIT-ENG-14-M-49, Air Force Institute of Technology Graduate School of Engineering and Management, Wright-Patterson Air Force Base, 2014.
5. P. Thompson, "Weak Models for Insider Threat Detection," *Proceedings of SPIE*, vol. 5403: "Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense," 2004, pp. 40–48.
6. T.A. Lewis, "An Artificial Neural Network-Based Decision Support System for Integrated Network Security," master's thesis no. AFIT-ENG-T-14-S-09, Air Force Institute of Technology Graduate School of Engineering and Management, Wright-Patterson Air Force Base, 2014.
7. C.J. Wood, "What Friends Are For: Collaborative Intelligence Analysis and Search," master's thesis, Naval Postgraduate School, 2014.
8. D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. New York: Cambridge University Press, 2010.
9. J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, 2004, pp. 5–53.
10. D.D. Lee and H.S. Seung, "Algorithms for Non-negative Matrix Factorization," in *Advances in Neural Information Processing Systems: The Proceedings of the 2000 Neural Information Processing Systems Conference*, T.K. Leen, T.G. Dietterich, and V. Tresp, eds., 2001, available at http://papers.nips.cc/book/advances-in-neural-information-processing-systems-13-2000.
11. V. Gadepally, J. Bolewski, D. Hook, D. Hutchinson, B. Miller, and J. Kepner, "Graphulo: Linear Algebra Graph Kernels for NoSQL Databases," 29th IEEE International Parallel and Distributed Processing Symposium Workshops, Hyderabad, India, 25–29 May 2015.
12. R. Lemos, "Survey Says Security Products Waste Our Time," *arstechnica*, 16 Jan. 2015, http://artechnica.com/security/2015/01/survey-says-security-products-waste-our-time/.
13. M. Suby and F. Dickson, "The 2015 $(ISC)^2$ [Information Systems Security Certification Consortium] Global Information Security Workforce Study," Frost and Sullivan White Papers, 16 April 2015, available at https://www.isc-2cares.org/IndustryResearch/GISWS/.
14. National Institute of Standards and Technology, National Vulnerability Database, available at https://nvd.nist.gov/.
15. J. O'Donovan and B. Smyth, "Trust in Recommender Systems," *Proceedings of the 10th International Conference on Intelligent User Interfaces*, 2005, pp. 167–174.
16. P. Avesani, P. Massa, and R. Tiella, "A Trust-Enhanced Recommender System Application: Moleskiing," *Proceedings of the 2005 ACM Symposium on Applied Computing*, 2005, pp. 1589–1593.

17. S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci, "Enhancing Privacy and Preserving Accuracy of a Distributed Collaborative Filtering," *Proceedings of the 2007 ACM Conference on Recommender Systems*, 2007, pp. 9–16.
18. V. Gadepally, B. Hancock, B. Kaiser, J. Kepner, P. Michaleas, M. Varia, and A. Yerukhimovich, "Improving the Veracity of Homeland Security Big Data through Computing on Masked Data," 2015 IEEE International Symposium on Technologies for Homeland Security, Waltham, Mass., 14–16 Apr. 2015.
19. E. Shen, M. Varia, R.K. Cunningham, and W.K. Vesey, "Cryptographically Secure Computation," *Computer*, vol. 48, no. 4, 2015, pp. 78–81.
20. A. Gunawardana and G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *Journal of Machine Learning Research*, vol. 10, 2009, pp. 2935–2962.

About the Authors

Vijay N. Gadepally is a technical staff member in Lincoln Laboratory's Secure Resilient Systems and Technology Group and a researcher at the MIT Computer Science and Artificial Intelligence Laboratory. His research is focused on big data and Internet of Things systems, machine learning, high-performance computing, advanced database technologies, and pattern recognition. Prior to joining Lincoln Laboratory in 2013, he worked as a postgraduate intern with the Raytheon Company, a visiting scholar with the Rensselaer Polytechnic Institute, and a student intern with the Indian Institute of Technology. He holds a bachelor's degree in electrical engineering from the Indian Institute of Technology in Kanpur, and master's and doctoral degrees in electrical and computer engineering from The Ohio State University. His doctoral dissertation on signal processing focused on developing mathematical models to accurately estimate and predict driver behavior for autonomous vehicle applications.

Braden J. Hancock is a computer science doctoral student at Stanford University and a former 2014 and 2015 summer intern in Lincoln Laboratory's Cyber Security and Information Sciences Division. His current research focus is on the development of machine learning techniques for automatically converting unstructured data into structured information for integration and inference tasks. Between his time at Stanford and as an undergraduate in mechanical engineering at Brigham Young University, he has received numerous awards, including the National Science Foundation (NSF) Graduate Research Fellowship, National Defense Science and Engineering Graduate Fellowship (declined to accept NSF fellowship), Phi Kappa Phi Marcus L. Urann Fellowship, Finch Family Fellowship (Stanford), Barry M. Goldwater Scholarship, American Society of Mechanical Engineers Kenneth Andrew Roe Scholarship, and American Institute of Aeronautics and Astronautics Vicki and George Muellner Scholarship. Prior to his internships at Lincoln Laboratory, he interned at the Johns Hopkins University Human Language Technology Center of Excellence and the Air Force Research Laboratory.

Kara B. Greenfield is a technical staff member in the Human Language Technology Group. Her work focuses on research in named-entity recognition, social network analysis, and visual analytics. Prior to joining Lincoln Laboratory, she interned at CA Technologies and Pegasystems, collaborated with HP Labs and the Hungarian Academy of Sciences, and served as a teaching assistant at Worcester Polytechnic Institute (WPI), where she won the Teaching Assistant of the Year Award from the Department of Mathematical Sciences. She holds bachelor's degrees in mathematics and computer science and a master's degree in industrial mathematics from WPI. Her master's research focused on utilizing crowdsourcing to develop a gold-standard-quality corpus for named-entity recognition. She is a member of two honor societies: Pi Mu Epsilon and Upsilon Pi Epsilon.

Joseph P. Campbell is the associate leader of Lincoln Laboratory's Human Language Technology Group, where he directs the group's research in speech, speaker, language, and dialect recognition; word and topic spotting; speech and audio enhancement; speech coding; text processing; natural language processing; machine translation of speech and text; information retrieval; extraction of entities, links, and events; cross-language information retrieval; multimedia recognition techniques, including both voice and face recognition for biometrics applications; advanced analytics for analyzing social networks on the basis of speech, text, video, and network communications and activities; and recommender systems. He specializes in the following for government applications: research, development, evaluation, and transfer of speaker recognition technologies; design of speaker recognition and biometrics evaluations; design of corpora to support those evaluations; and development and evaluation of biometrics technologies and systems. He joined Lincoln Laboratory in 2001 as a senior staff member after serving 22 years at the National Security Agency. He was an IEEE Distinguished Lecturer and is an IEEE Fellow. He earned bachelor's, master's, and doctoral degrees in electrical engineering from Rensselaer Polytechnic Institute, The Johns Hopkins University, and Oklahoma State University, respectively.

William M. Campbell is a senior technical staff member of the Human Language Technology Group. He provides leadership and technical contributions in the areas of speech processing, machine learning, and social networks. His speech processing work has resulted in advances in speaker and language recognition, including the development of algorithms that have been widely cited in published papers and implemented. He has made numerous contributions in social network graph analysis as it relates to simulation of social networks; machine learning involving social networks; and construction of networks from multimedia content. Prior to joining Lincoln Laboratory in 2002, he worked on speech processing and communication systems at Motorola. An active contributor to the speech and machine-learning research community, he has served as reviewer and scientific committee member for several conferences: IEEE Odyssey: The Speaker and Language Recognition Workshop; Annual Conference on Neural Information Processing Systems; International Conference on Acoustics, Speech, and Signal Processing; INTERSPEECH; and IEEE Spoken Language Technology Workshop. He is the author of more than 100 peer-reviewed papers, including multiple book chapters; a recipient of the Motorola Distinguished Innovator Award; the holder of 14 patents; and a senior member of the IEEE. He received three bachelor's degrees—in computer science, electrical engineering, and mathematics—from South Dakota School of Mines and Technology and master's and doctoral degrees in applied mathematics from Cornell University.

Albert I. Reuther is the assistant group leader of the Secure Resilient Systems and Technology Group, where he leads research teams in graph processing and analytics, high-performance parallel and distributed computing, machine learning, and novel computer architectures. He joined Lincoln Laboratory in 2001. His publications include a number of papers and book chapters on interactive, on-demand supercomputing, dynamic computational and database prototyping environments, parallel and distributed signal processing, and supercomputing scheduling algorithms. He received the 2005 Eaton Award in Design Excellence from Purdue University's School of Electrical and Computer Engineering, was an Intel Foundation Graduate Fellowship recipient, and is a member of the IEEE and Association for Computing Machinery (ACM) professional societies. He earned a dual bachelor's degree in computer and electrical engineering, a master's degree in electrical engineering, and a doctoral degree in electrical and computer engineering, all from Purdue University, and a master's of business administration degree from the Collège des Ingénieurs in Paris, France, and Stuttgart, Germany.