Towards a Computational Model of Human Opinion Dynamics in Response to Real-World Events

Kallirroi Georgila and **David V. Pynadath**
Institute for Creative Technologies, University of Southern California
12015 Waterfront Drive, Playa Vista, CA 90094, USA

Abstract

Accurate multiagent social simulation requires a computational model of how people incorporate their observations of real-world events into their beliefs about the state of their world. Current methods for creating such agent-based models typically rely on manual input that can be both burdensome and subjective. In this investigation, we instead pursue automated methods that can translate available data into the desired computational models. For this purpose, we use a corpus of real-world events in combination with longitudinal public opinion polls on a variety of opinion issues. We perform two experiments using automated methods taken from the literature. In our first experiment, we train maximum entropy classifiers to model changes in opinion scores as a function of real-world events. We measure and analyze the accuracy of our learned classifiers by comparing the opinion scores they generate against the opinion scores occurring in a held-out subset of our corpus. In our second experiment, we learn Bayesian networks to capture the same function. We then compare the dependency structures induced by the two methods to identify the event features that have the most significant effect on changes in public opinion.

Introduction

Social scientists have increasingly turned to multiagent social simulation as a means for representing and analyzing human behavior on a large scale (Miller and Page 2007). Agent-based modeling provides an appealingly generative model of the decision-making of the human individuals and groups to be analyzed (Davidsson 2002). As a result, there is a growing number of successful applications of multiagent systems to model, simulate, and analyze real-world social scenarios (Alam and Parunak 2014; Kennedy, Agarwal, and Yang 2014).

These applications face an enormous hurdle in accurately capturing all of the various aspects of human decision-making. One particularly challenging aspect is the process by which people form their beliefs about the world and how they update those beliefs over time. There is a large body of work on how such beliefs spread through social networks, where one individual's opinions can influence his or her neighbors (Scott and Carrington 2011). Researchers have combined agent-based modeling with complex networks to simulate how public opinion formation is shaped by such local interactions, as well as by environmental factors (e.g., topography, demographics, etc.) (Suo and Chen 2008).

However, while social influence and environmental features certainly play a role in opinion formation, day-to-day events have a strong impact as well. For example, it is unlikely that someone's opinion about national security will improve after an IED detonation on their street, regardless of the opinions of their social network neighbors. In this work, we seek to capture this complementary influence of local events on the belief formation process. In particular, we investigate methods for constructing an agent-based model of belief update that can capture how people's opinions evolve in response to exogenous events in their environment.

While there exist multiagent social simulation frameworks that provide architectural mechanisms for capturing human belief formation (Luke et al. 2005; Pynadath and Marsella 2005), these frameworks also require manual labor to create the domain modeling input. In addition to imposing a considerable burden, this authoring process is also likely to introduce inaccuracy and subjectivity to the resulting models. We instead seek a data-driven approach to constructing the desired agent-based models of opinion formation and dynamics. We can exploit novel data sources like the Global Database on Events, Location, and Tone (GDELT), which collects events from international news sources on a daily basis, recording the who, what, and where of the events in a computer-readable database format (Leetaru and Schrodt 2013). We use GDELT as our source for local events, in conjunction with longitudinal survey results from Afghanistan as our source for public opinions that reflect the civilian population's beliefs (Hopkins 2013). By aligning the time series of events with the evolution of survey responses, we can explore the effectiveness of candidate models of the belief update process.

We start by viewing this alignment as a classification problem. More specifically, we seek to learn the distribution over changes in survey results across each region of Afghanistan as a function of the news events in that region. Maximum entropy (MaxEnt) classifiers have proven successful in learning such distributions in natural language processing and other domains (Kazama and Tsujii 2003). Therefore, we first apply a MaxEnt classifier to learn a be-

lief update function from our data and evaluate the accuracy of the result against the actual survey responses. By examining the learned classifiers, we gain insight into the event variables that have the most significant impact on the belief changes observed from the survey results.

A classification approach allows us to learn a functional mapping from event variables to opinion variables, but it does not exploit the semantics of these opinions as human beliefs about the state of the world. Researchers have hypothesized that Bayesian networks provide a framework for modeling how humans infer causality (Pearl 2000). Therefore, while Bayesian networks can also classify belief updates in the same functional manner as MaxEnt approaches, they also support learning algorithms that can generate a probabilistic model of the world that can support more general belief queries. We apply such learning algorithms to our same data sets to learn a Bayesian network representation of the joint distribution over events and survey results.

The result is a data-driven approach to modeling human inferences about the world we live in both as a MaxEnt classifier and as a Bayesian network. We compare the dependency structures generated by both methods to identify the event features that play the most significant role in the observed belief changes. Not surprisingly, given our completely domain knowledge-free learning approach, the accuracy of our learned belief update models does not compare to the results achieved on more traditional machine learning domains. However, for some survey questions the learned models do show a marked improvement over baseline approaches, demonstrating the promise of applying such agent learning algorithms to modeling human belief formation and dynamics. Furthermore, the Bayesian network formalism provides a rich modeling language for exploiting domain knowledge beyond what we used in the current investigation's purely data-driven approach. This line of investigation thus provides important first steps towards a model of human belief update for social simulation that fully exploits the computational leverage provided by advances in multiagent research (Wellman 2014).

Data

We use the GDELT database as a source of real-world events (publicly available at http://www.gdeltproject.org). The GDELT database contains events that have taken place all over the world in the last 35 years. For our experiments we use data from 2007 until 2013, and focus only on events that took place in Afghanistan. The second source of our data is results of surveys that the Asian Foundation conducts in Afghanistan on a yearly basis (every summer), starting in 2006.[1] These survey results are available per Afghanistan region and can be freely downloaded from http://www.asianfoundation.org. We use the results of their surveys from 2006 to 2013. We associate these survey scores with events in the GDELT database. For example, an event that took place in November 2006 at the western region of Afghanistan will be associated with the survey results of summer 2007 for that particular region. Note that in Afghanistan there are 8 major regions (Central-Kabul, East, South-East, South-West, West, North-East, Central-Hazarjat, and North-West). The survey scores of each year are associated with events up to 1 year in the past.

For our experiments, we use the following survey questions:

- Q3: Overall, based on your own experience, do you think things in Afghanistan today are going in the right direction, or do you think they are going in the wrong direction? (right direction, wrong direction, don't know)
- Q6: Overall, for you and your family, which of these periods was the best economically? (most recent period after the Taliban, Taliban period, no difference, don't know)
- Q18: How often do you fear for your own personal safety or security, or for that of your family these days? (always, rarely, sometimes, often, never, don't know)
- Q9a: Would you rate... The availability of clean drinking water as good or bad in your area? (very good, quite good, quite bad, very bad, don't know)
- Q9b: Would you rate... The availability of water for irrigation as good or bad in your area? (very good, quite good, quite bad, very bad, don't know)
- Q9c: Would you rate... The supply of electricity as good or bad in your area? (very good, quite good, quite bad, very bad, don't know)
- Q9d: Would you rate... The availability of clinics and hospitals as good or bad in your area? (very good, quite good, quite bad, very bad, don't know)
- Q9e: Would you rate... The availability of medicine as good or bad in your area? (very good, quite good, quite bad, very bad, don't know)
- Q9f: Would you rate... The availability of education for children as good or bad in your area? (very good, quite good, quite bad, very bad, don't know)

We convert the above Likert scale to a numerical scale from 0 to 2 (for questions Q3 and Q6), and to a numerical scale from 0 to 4 (for questions Q18, Q9a, Q9b, Q9c, Q9d, Q9e, and Q9f), where 2 or 4 corresponds to the most positive response and 0 to the most negative response. We assign "don't know" responses to the middle of the scale. Then, taking into account the probability distribution of these scores (how many people chose each possible response), we come up with a single score for each survey question, the value of which ranges from 0 to 1 (this is a real number; 1 means most positive and 0 means most negative opinion). For example, for question Q3, if 45% of the population selected "right direction", 35% selected "don't know", and 20% selected "wrong direction" then the score would be (0.45*2+0.35*1+0.2*0)/2=0.625. In practice these scores range from 0.1 to 0.9 depending on the survey question. Mostly they vary between 0.4 and 0.8.

[1] This is why we only use GDELT events after 2007. In our experiments, each event has as attributes (features) the survey scores of the previous year (see the methodology section). Thus we start from year 2007 so that we can use the scores of 2006 as previous-year opinion scores.

In each event of the GDELT database there is an actor Actor1 who performs an action (Action) that affects Actor2. Each event has the following attributes related to Actor1:

- Actor1Code: includes geographic, class, ethnic, religious, and type information.
- Actor1Name: the actual name of Actor1.
- Actor1CountryCode: a 3-character code for the country of Actor1.
- Actor1KnownGroupCode: denotes if Actor1 is a known IGO/NGO/rebel organization (e.g., United Nations, World Bank, al-Qaeda) or not.
- Actor1EthnicCode: the ethnic affiliation of Actor1.
- Actor1Religion1Code, Actor1Religion2Code: the religious affiliation of Actor1 (there could be more than one religion code).
- Actor1Type1Code, Actor1Type2Code, Actor1Type3Code: the specific role of Actor1, e.g., police forces, government, military, political opposition, rebels, etc. (there could be more than one type code).

There are also codes that provide geographical information, such as Actor1Geo_Type (e.g., country, US state, US city, etc.), Actor1Geo_Fullname (the full name of the location of Actor1), Actor1Geo_CountryCode (a 2-character country code), Actor1Geo_ADM1Code (each country is divided in regions, and this code is associated with a region).

The same fields are used for Actor2 and Action. There are also attributes related to the action (event):

- IsRootEvent: this code is a rough proxy for the importance of an event.
- EventCode, EventBaseCode, EventRootCode: these are all codes for the type of the event, e.g., make optimistic comment, acknowledge or claim responsibility, appeal for humanitarian aid, etc., and they are part of a taxonomy of event codes.
- QuadClass: it can take 4 values (verbal cooperation, material cooperation, verbal conflict, and material conflict).
- GoldsteinScale: a numeric score from -10 to 10 capturing the potential impact that this type of event could have on the stability of a country.
- NumMentions: the total number of mentions of the event across all source documents (indicates the importance of the event).
- NumSources: the total number of information sources containing one or more mentions of the event (indicates the importance of the event).
- NumArticles: the total number of source documents containing one or more mentions of the event (indicates the importance of the event).
- AvgTone: the average "tone" of all documents mentioning the event. This score ranges from -100 (extremely negative) to +100 (extremely positive). Usually values vary between -10 and +10.

Finally, there are attributes that provide information about the date that the event took place, the geographic latitude and longitude, a unique identification code for each event, etc.

Methodology

We use the GDELT event data with the corresponding survey scores in order to train 9 MaxEnt classifiers[2], one per survey question (Q3, Q6, Q18, Q9a, Q9b, Q9c, Q9d, Q9e, and Q9f). The goal of each classifier is to predict the survey score for a new unseen event.

For each classifier we use all the event attributes related to Actor1, Actor2, and Action, e.g., Actor1Code, Actor1Name, GoldsteinScale, NumMentions, etc. (37 attributes in total). For each classifier and each event in a particular region, we also use as additional features the previous-year survey scores (for that particular region) for all survey questions (9 additional attributes). From now on these additional features will be referred to as Q3_prev, Q6_prev, Q18_prev, Q9a_prev, Q9b_prev, Q9c_prev, Q9d_prev, Q9e_prev, and Q9f_prev. Thus for the Q3 classifier, the features used for classification are the 37 GDELT attributes plus the 9 additional features, and the output of the classifier is the value of Q3; and likewise for the rest of the survey questions.

As mentioned in the data section, the survey scores are real numbers ranging from 0 to 1. To perform classification we cluster these scores into 10 bins (or classes)[3]: scores in the interval (0,0.1] correspond to bin C1, (0.1,0.2] to C2, (0.2,0.3] to C3, (0.3,0.4] to C4, (0.4,0.5] to C5, (0.5,0.6] to C6, (0.6,0.7] to C7, (0.7,0.8] to C8, (0.8,0.9] to C9, and (0.9,1] to C10.

We split our data set as follows: we use events from years 2007–2010 as training data (144000 events in total) and events from years 2011–2013 as test data (108000 events in total). This is a realistic way of splitting the data, as the idea is to build models using data from the past and apply them to future events. We train our models on the training data and test them on the test data. Because we have 9 survey questions, we create 9 training sets and 9 test sets. These data sets are all similar (they use the same features, as mentioned above) except for the value of the class to be predicted. For the Q3 data sets the class to be predicted corresponds to the survey scores for question Q3, for the Q6 data sets the class to be predicted corresponds to the scores for question Q6, etc.

To evaluate our classifiers we use the following metrics. To understand how these metrics work let us assume that in our test data 40% of the events are associated with class C5 (the actual survey score is always 0.49) and 60% of the events are associated with C6 (the actual survey score is always 0.54). Let us also assume that for all events our model always predicts C6 with a probability 0.7 and C5 with a probability 0.3.

- **Accuracy:** This is equal to the number of classes that have been predicted correctly divided by the total number of events. For the above example, the accuracy is 0.6 = 60% (the model always predicts C6).
- **Weighted Accuracy:** Weighted accuracy (or expected accuracy) shows the percentage of the time that the model

[2]We use a C++ library from http://www.nactem.ac.uk/tsuruoka/maxent/ (Kazama and Tsujii 2003).

[3]From now on we will use the terms "bin" and "class" interchangeably.

would choose the same class as the actual class in the test data if we ran the model for a large number of times through the same events in the test data, each time generating a class according to the distribution proposed by the model. For the above example, the weighted accuracy is 0.6*0.7+0.4*0.3 = 0.54 = 54%. This metric is also known as "predicted probability" (Zukerman and Albrecht 2001).

- **Distance:** Obviously accuracy is a very strict metric and depends a lot on how the classification bins have been defined. For the above example, given that the actual survey score is 0.49 and our model predicts C6, this will be classified as wrong even though the value 0.49 is very close to the bin C6. For this reason, we use the distance metric to measure how close to the actual survey score the mean value of the predicted bin is. For the above example, the distance would be equal to 0.6*(0.55-0.54)+0.4*(0.55-0.49) = 0.03 (the model always predicts C6 and 0.55 is the mean value of bin C6).
- **Weighted Distance:** Weighted distance (or expected distance) bears the same relation to distance as weighted accuracy to accuracy. In other words, for each event, this metric does not take into account only the predicted class with the highest probability but all predicted classes (even the ones with lower probabilities). For the above example, the weighted distance would be equal to 0.6*[0.7*(0.55-0.54)+0.3*(0.54-0.45)] + 0.4*[0.7*(0.55-0.49)+ 0.3*(0.49-0.45)] = 0.042 (0.55 is the mean value of bin C6 and 0.45 is the mean value of bin C5).

The reason for using the weighted versions of all metrics is that we want to measure the effect of the full probability distribution that the model uses for its predictions, not just the effect of the predictions with the highest probability. As we will see in the results section, the majority baseline, that we compare our MaxEnt model against, does not perform well on these metrics when the distribution of classes in the training data is very different from the distribution of classes in the test data. However, if it just happens that the majority class in the training data is the same as the majority class in the test data (even if the full distributions differ), the majority baseline will do well on the non-weighted metrics.

We also use the above training data sets to learn 9 Bayesian networks with the Microsoft WinMine Toolkit available at http://research.microsoft.com/en-us/um/people/dmax/WinMine/tooldoc.htm (one network per survey question). We report dependencies between the survey result and the event attributes.

Results

Below we report our classification results for all survey questions and metrics. We compare with a majority baseline, which always predicts the class which has the highest number of occurrences in the training data. For weighted accuracy and weighted distance, the majority baseline makes predictions based on the probability distribution of classes in the training data. In Table 1 we can see the results for accuracy, weighted accuracy, distance, and weighted distance, for the MaxEnt models and the majority baseline models, and for all survey questions.

The MaxEnt model performs better for questions Q6, Q9c, Q9e, and Q9f. The majority baseline performs better for questions Q18, Q9a, and Q9b. The two models (MaxEnt and majority baseline) perform similarly for questions Q3 and Q9d (one is better than the other depending on the metric used). It is encouraging that the distance values are always very low even when our model does not perform very well in terms of accuracy. The distance values are less than 0.1 or very close to 0.1, which means that even when our model makes wrong predictions, these predictions are not far off the true values.

We also calculated precision, recall, weighted precision, and weighted recall scores per class (C1-C10). Due to space constraints we cannot report detailed results, but the bottom line is that our models have a more consistent performance across all classes, whereas the majority baseline, as expected, performs well only when the distribution of classes in the training data matches the distribution of classes in the test data. However, because there is not much variation in the survey scores over the years, the majority baseline ends up being a strong baseline. The GDELT data are very noisy and sparse. Many of the attributes are not available most of the time, which makes the classification problem harder. So in that sense it is encouraging that our MaxEnt model performs comparably with such a strong baseline.

In Table 2 we can see the dependencies between the survey score variables and the event attributes, as a result of training Bayesian networks on the training data for each survey question. The structures of the networks vary depending on the survey question. The MaxEnt models appear to be more consistent across survey questions. The most significant MaxEnt features (larger feature weights) are NumMentions, NumSources, NumArticles, GoldsteinScale, QuadClass, AvgTone, Actor1Code, Actor2Code, all the "Event" and "ADM1" codes, and the survey scores of previous years.

Our MaxEnt models demonstrate very encouraging success given that we do not use any domain-specific knowledge in learning them. The fact that one year's survey responses significantly depend on the previous year's may seem obvious, given the relative stability of opinions. However, it is a nice validation of our method that it is able to identify this dependency, out of all of the other equally weighted hypotheses, without any prior knowledge to bias its search.

Table 2 contains additional examples of how our methodology can validate the degree to which certain intuitive relationships hold. We see that the number of different sources that cover an event (NumSources) is in the envelope for all but one of our target questions. Such a strong dependency might exist because the more sources that cover an event, the greater the penetration the coverage will have within the populace. In contrast, the number of mentions (NumMentions) and articles (NumArticles) for an event appear only four times each, and only once very strongly. This result suggests that the same source repeating coverage of the same event has a lesser impact, as it is repeating the coverage to largely the same audience.

Given that one's perceptions about the availability of wa-

	Q3		Q6		Q18		Q9a		Q9b	
	MaxEnt	Majority	MaxEnt	Majority	MaxEnt	Majority	MaxEnt	Majority	MaxEnt	Majority
Accuracy	43.93	59.33	36.71	24.10	43.21	55.78	13.12	51.13	26.51	36.85
Wgt accuracy	40.48	34.76	31.87	18.20	42.05	43.21	14.71	30.84	30.14	42.87
Distance	0.06	0.04	0.11	0.11	0.07	0.06	0.11	0.06	0.06	0.04
Wgt distance	0.07	0.08	0.12	0.16	0.07	0.07	0.11	0.10	0.06	0.06
	Q9c		Q9d		Q9e		Q9f			
	MaxEnt	Majority	MaxEnt	Majority	MaxEnt	Majority	MaxEnt	Majority		
Accuracy	48.48	25.63	21.63	10.04	51.83	35.12	43.07	41.70		
Wgt accuracy	46.10	20.16	22.45	29.43	49.22	37.91	46.77	34.73		
Distance	0.07	0.14	0.08	0.11	0.06	0.09	0.08	0.07		
Wgt distance	0.07	0.17	0.08	0.08	0.07	0.08	0.08	0.11		

Table 1: Results for accuracy, weighted (wgt) accuracy, distance, and weighted (wgt) distance, for the MaxEnt models and the majority baseline models

Q3: Actor1Type1Code (P), Actor1CountryCode (P, S), Actor1Geo_Type (P, S), Actor2CountryCode (P), Actor2Geo_Type (P, S), ActionGeo_Type (P, S), ActionGeo_ADM1Code (P, S), EventRootCode (P), EventCode (P), AvgTone (P, S), NumArticles (P, S), NumMentions (P), Q9e_prev (P, S), NumSources (C), Q3_prev (C, S), Q6_prev (C, S), Q18_prev (C, S), Q9a_prev (C, S), Q9b_prev (C, S), Q9c_prev (C, S), Q9f_prev (C, S)
Q6: Q3_prev (P, S), Q6_prev (P, S), Q18_prev (P, S), Q9a_prev (P, S), Actor1Type2Code (C), NumSources (C), Q9f_prev (C, S)
Q18: Actor2KnownGroup (P), EventRootCode (P), EventBaseCode (P), EventCode (P), QuadClass (P), GoldsteinScale (P, S), AvgTone (P, S), Actor1Type1Code (C), Actor1Geo_Type (C, S), ActionGeo_Type (C, S), ActionGeo_ADM1Code (C, S), IsRootEvent (C), NumMentions (C), Q3_prev (C, S), Q6_prev (C, S), Q18_prev (C, S), Q9a_prev (C, S), Q9b_prev (C, S), Q9c_prev (C, S), Q9e_prev (C)
Q9a: Actor1CountryCode (P), Actor1Geo_CountryCode (P), Actor1Geo_Type (P), Actor2CountryCode (P), Actor2Geo_Type (P, S), ActionGeo_Type (P), ActionGeo_ADM1Code (P, S), EventRootCode (P), EventCode (P), NumArticles (P), NumMentions (P), AvgTone (P, S), Q9e_prev (P, S), Actor1Type2Code (C), NumSources (C, S), Q3_prev (C, S), Q6_prev (C, S), Q18_prev (C, S), Q9a_prev (C, S), Q9b_prev (C, S), Q9c_prev (C, S)
Q9b: Actor1CountryCode (P), Actor1Geo_CountryCode (P), Actor1Geo_Type (P, S), Actor2CountryCode (P), Actor2Geo_CountryCode (P, S), Actor2Geo_Type (P, S), ActionGeo_Type (P), ActionGeo_ADM1Code (P, S), EventRootCode (P), EventCode (P), QuadClass (P), NumArticles (P), AvgTone (P, S), Q9e_prev (P, S), NumSources (C, S), Q3_prev (C, S), Q6_prev (C, S), Q18_prev (C, S), Q9a_prev (C, S), Q9b_prev (C, S), Q9c_prev (C)
Q9c: Q3_prev (P, S), Q6_prev (P, S), Q18_prev (P, S), Q9a_prev (P, S), Q9e_prev (P, S), Actor1Type2Code (C), NumSources (C), Q9f_prev (C, S)
Q9d: Actor2Type3Code (P), EventRootCode (P), EventCode (P), NumArticles (P), AvgTone (P), Q9e_prev (P), ActionGeo_Type (C, S), ActionGeo_ADM1Code (C), IsRootEvent (C), NumSources (C), Q3_prev (C, S), Q6_prev (C), Q18_prev (C, S), Q9a_prev (C, S), Q9b_prev (C, S), Q9c_prev (C, S)
Q9e: EventRootCode (P), EventBaseCode (P), EventCode (P), AvgTone (P), Q9a_prev (P, S), Actor2KnownGroup (C), Actor2Religion1Code (C), ActionGeo_Type (C, S), ActionGeo_ADM1Code (C, S), IsRootEvent (C), NumMentions (C), NumSources (C), Q3_prev (C, S), Q18_prev (C, S), Q9d_prev (C, S), Q9e_prev (C, S)
Q9f: Actor2KnownGroup (P), EventRootCode (P), EventBaseCode (P), EventCode (P), GoldsteinScale (P, S), AvgTone (P, S), Actor2Geo_Type (C), ActionGeo_Type (C, S), ActionGeo_ADM1Code (C, S), IsRootEvent (C), NumSources (C, S), Q3_prev (C, S), Q6_prev (C, S), Q9a_prev (C, S), Q9b_prev (C, S), Q9c_prev (C, S), Q9e_prev (C, S)

Table 2: Dependencies per survey question (S: very strong dependency, P/C: parent/child of survey score variable)

ter, medicine, education, etc. would influence one's assessment of the direction the country is going, it is not surprising to see many interdependencies among our question variables in Table 2. Q18 appears within the envelope of all of our question variables except one, suggesting that the availability of basic services is intertwined with our level of safety. Of course, this dependency could indicate that a lack of security makes it impossible to provide basic services, or it could indicate a dependency that exists in only our *perceptions*. The one question where Q18 does *not* appear in the envelope (Q9f) may be more informative in this regard, as the dependency between security and education is apparently not as strong as with the other services. These differences in dependencies identify interesting avenues of further investigation for more accurately modeling public perceptions.

Conclusion

The encouraging results from our almost naive application of machine learning methods suggest that we can achieve even better results with some next steps that further exploit the Bayesian network model. In particular, Table 2 indicates dependencies among variables that may exist for only some of the possible values within those variables' domains. For example, the various event code features appear in the envelope for Q18, but we would expect that certain event types (e.g., "peacekeeping", "violate ceasefire") would

have a much stronger effect than others (e.g., "discuss by telephone", "engage in symbolic act"). In such cases, the Bayesian network would exhibit *context-specific independence* (Boutilier et al. 1996) that we could potentially identify and exploit in our model.

As already mentioned, the sparseness of the data was an obvious obstacle to learning accurate models. In some cases, it may be possible to remove features without losing all of their information. For example, we currently include the region of an event in our joint distribution, which leads to region-specific components within the learned model. We could alternatively remove the region, while still aligning only the events that occur within a region with the survey results from that region. Through this method, we would arrive at a model that represents a more "generic" Afghan's belief update. This may result in a loss of accuracy if there are differences in opinion formation psychology across the different regions. Even if so, creating this region-independent model will still provide a quantitative method for analyzing and understanding these differences. Another idea is to consider temporal interdependencies between events, e.g., by using maximum entropy Markov models or dynamic Bayesian networks.

While it is certainly desirable to have a computational model capable of forecasting survey responses, the model would be more useful if it could inform policy makers as well. To that end, the model must support counterfactual reasoning, so that a decision-maker can profile the potential outcomes of the candidate policies under consideration. Fortunately, one of the most successful uses of Bayesian networks is for exactly that kind of counterfactual reasoning (Pearl 2000). For example, a policy maker can use the learned Bayesian network, choose a set of events to perform (e.g., "provide economic aid" vs. "provide humanitarian aid"), and query the desired opinion variables to project the effects of those events on public opinion.

By translating real-world event and opinion data into an agent-based belief model like a Bayesian network, we gain the leverage of decades of research and algorithm development. We can capture a richer dependency structure than is possible with simpler classification frameworks, while still retaining a graphical representation that lends itself to human understanding. Furthermore, the use of a probabilistic belief representation opens the door for decision-theoretic behavior generation as well. Our methodology thus provides a path towards fully exploiting the expressivity and reasoning power of decision-theoretic multiagent systems in the context of data-driven social simulation.

We therefore believe that automated modeling of opinion formation is an important challenge problem for AI researchers and that this paper's results illustrate some of the progress that can be made.

Acknowledgments

The effort described here is supported by the U.S. Army. Any opinion, content or information presented does not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Alam, S. J., and Parunak, H. V. D., eds. 2014. *Multi-Agent-Based Simulation XIV*. Springer.

Boutilier, C.; Friedman, N.; Goldszmidt, M.; and Koller, D. 1996. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, 115–123.

Davidsson, P. 2002. Agent based social simulation: A computer science view. *Journal of artificial societies and social simulation* 5(1).

Hopkins, N., ed. 2013. *Afghanistan in 2013: A Survey of the Afghan People*. The Asia Foundation.

Kazama, J., and Tsujii, J. 2003. Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 137–144.

Kennedy, W. G.; Agarwal, N.; and Yang, S. J., eds. 2014. *Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer.

Leetaru, K., and Schrodt, P. A. 2013. GDELT: Global data on events, location, and tone, 1979–2012. In *Proceedings of the International Studies Association Annual Conference*, volume 2.

Luke, S.; Cioffi-Revilla, C.; Panait, L.; Sullivan, K.; and Balan, G. 2005. MASON: A multiagent simulation environment. *Simulation* 81(7):517–527.

Miller, J. H., and Page, S. E. 2007. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton University Press.

Pearl, J. 2000. *Causality: models, reasoning and inference*, volume 29. Cambridge University Press.

Pynadath, D. V., and Marsella, S. C. 2005. PsychSim: Modeling theory of mind with decision-theoretic agents. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1181–1186.

Scott, J., and Carrington, P. J. 2011. *The SAGE handbook of social network analysis*. SAGE publications.

Suo, S., and Chen, Y. 2008. The dynamics of public opinion in complex networks. *Journal of Artificial Societies and Social Simulation* 11(4):2.

Wellman, M. P. 2014. Putting the *agent* in agent-based modeling. Edited transcript of a talk presented at the *International Conference on Autonomous Agents and Multi-Agent Systems*.

Zukerman, I., and Albrecht, D. W. 2001. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction* 11(1–2):5–18.