Two Applications of Statistical Relational Learning: Fake News Detection and Congress Voting Patterns

Hao, Q; Peng, T

I. Introduction

1. Relational machine learning,

Different from individual traits, relational data capture information about the relations individuals have. Relational data describe the property of at least a pair of individual entities. The relations are usually represented with named links. Relational (machine) learning is the kind of method that trains models to predict the existence of the named links between entities.

2. Learning Algorithm

For a review of different kinds of statistical relational learning (SRL) algorithms, please see Nickel (2016). The current study specifically applies the Additive Relational Effects (ARE) model developed by Nickel, Jiang and Tresp (2014) in two empirical endeavors: fake news detection and congress voting pattern detection. Nickel's (2014) learning algorithm consists of two parts: link prediction based on entities' individual features ($\boldsymbol{R} \times_1 A \times_2 A$) and link prediction based on entities' indirect paths ($\boldsymbol{M} \times_3 W$). The effects of the two parts are additive and the eventual prediction is in the form of $\widehat{\boldsymbol{X}} = \boldsymbol{R} \times_1 A \times_2 A + \boldsymbol{M} \times_3 W$. (Equation 1)

In a nut shell, the learning algorithm tries to learn three things, A, **R** and W. Matrix A contains each entity's latent features, which can be used to predict how likely one entity (with certain latent features) is to have a certain relationship with other entities (with certain latent features); second, tensor **R** contains information on how much each pair of latent features predict a certain link between pairs of entities having this pair of latent features; third, matrix W captures how likely the entities with a specified indirect relationship with each other are to have a certain direct relationship. Tensor **M** contains descriptive information on whether each pair of entities have a certain indirect link. Tensor **X** contains the descriptive information on whether each pair of entities have a certain direct link. Both X and M can be directly constructed from the dataset.

The way of learning these three things is through tensor factorization. Assume there is a latent feature matrix A, whose rows are latent feature vectors of each entity, and there is a weight matrix W, whose ij-th element is the weight of the jth indirect relationship in predicting the ith direct relation between entities. Then one can factorize **X** using A and W, writing **X** as a function of (products and sums of) A and W according to equation 1 and finding the A and W that best reproduces **X**. Nickel et al. (2014) gave an algorithm that factorizes **X** using A, W, and **R**, The latent features in A and weights in R and W are what have been *learned* from the training set (**X**) about the entities and their relations.

The current paper will apply the algorithm to two datasets: fake news dataset and congress votes dataset. It will be shown that even without an explicit coding scheme of what fake news actually is, the algorithm is able to pick up fake news in the testing set with certain level of accuracy. It will also be demonstrated that the observed relationships between entities can be reduced to latent features and indirect relationships (through factorization). Attempts will be made to interpret the latent features and patterns that drive the formation of the observed relationships.

II. Application to Fake News Detection

In this age of social media, fake news has become an important social phenomenon. From the public's perception of to individuals' decision making on social and political issues, fake news, or mis-information, has found its way to exert its influences. It is now more appropriate than ever to study fake news. Heuristics such as the source of the tweet could be used as classifiers for twitter news (e.g. Rubin, Conroy, Chen & Cornwell, 2016). This study takes a different approach: to detect fake news based on their contents. This section will apply the ARE model to do content-aware detection of fake news on twitter. This algorithm will represent each factual statement as stating a relation between two entities. A network of factual statements can be built to represent multiple pieces of news, where nodes are entities' names and edges are relational words connecting the entities in people's tweets. Under this representation, true news, meaning that the stated relational word actually does exist between the two entities, is equivalent to the existence of a link in the network; fake news is equivalent to a non-existent link in the network. The task of fake news prediction becomes a problem of link prediction on a network. The study will obtain some tweets, construct the network representing the stated facts in the tweets corpora and use a machine learning algorithm to do link prediction on the constructed network of factual statements.

1. Data

Any nouns or phrases that serve as a noun in a sentence are noun phrases (NP), or, entities. The word that states the relation between the two entities is called a relational word. For example, in "vaccination causes autism", both 'vaccination' and 'autism' are NPs and 'cause' is the relational word. Data obtaining started with

1.1) selecting a small set of key words of topics that potentially harbor fake news (i.e. "climate change, global warming, election, GMO, illegal votes, mass shooting, Obama, Ruusia, Pope, vaccination"). Then,

1.2) twitter text data on the selected topic words from the area of Michigan state during early February and mid-March were obtained through the twitter API.

1.3) The most frequently used relational words in the tweets were extracted. (i.e. "agree", "disagree", "cause", "is", "faked/manipulated", "ruin", "deny", "kill", "reduce", "damage", "is/are in").

1.4) The data obtained from step 1.2 went through some initial cleaning (e.g. removing the http link at the end, removing emojis, converting text encoding format etc.) before they were used for information extraction.

The dataset consisted of approximately 200,000 tweets in total, half of them from February and half from March, 2018 in Michigan. the February dataset was used as training set and the March dataset testing set.

2. Data Preparation

The factual relations that people tweeted about were extracted from the tweets.

2.1) First, the researcher extracted all the potential relational triplets (entity, relational, entity) from the twitter text data. The information extraction went through five steps (sentence segmentation, tokenization, part of speech tagging, entity recognition and relation recognition). All the five steps were done with the nltk toolkit in python. The first two steps were each achieved with a few lines of code.

To do part of speech (POS) tagging, the current study utilized the trained taggers stored in nltk toolkit. Nltk stored the most probable tags for most frequently used words and assigned the same tag to the same word in all cases. To improve the performance of this baseline tagging model, POS tagging in this research was done by specifying backoff taggers, which combined the results of a bigram tagger, unigram tagger and a default tagger (Natural Language Processing with Python, p. 227). Bigram taggers were trained model in the nltk package that stored each words' most probable tag given the word appearing before it. The backoff lookup tagging process would 1) first try tagging the tokens with bigram tagger, 2) if the bigram tagger was unable to find a tag for the token, try the unigram tagger, and 3) if the unigram tagger was also unable to find a tag, use a default tagger stored for that word. For this research, the researcher went through the tags of each sentence containing one of the 11 relational words of interest and manually confirmed that the tags were done correctly. Incorrect ones were manually corrected. With larger datasets, this process may become intractable. However, with large datasets the tolerance for noises may also increase. It remains an empirical question whether and how much the tagging inaccuracy affects the research results.

For entity recognition, a basic tag-based parsing algorithm was used. The researcher went through the text and looked at the noun phrases that needed to be extracted, summarized their patterns of existence in terms of tagging patterns and then defined a grammar using regular expression to extract the phrases that had the defined pattern. In this case, the researcher found the following patterns to be effective: “one or no determinant, one or more adjective, one noun”, “one or more nouns”, “noun, plural noun”. After the extraction, the researcher read through the extracted noun phrases and deleted the ones that did not make sense. Those falsely parsed cases that made no sense could be due to failing to define the appropriate grammar, applying the grammar on complex sentences or falsely tagged sentences. Sometimes, it could also be that the original tweet text was grammatically incorrect or unconventional. In this study, these cases were all deleted for both the training and testing set. Some spelling tricks were interpreted and corrected by human coding (e.g. bc = because, ppl = people, dems = democrats, Rs = republicans etc).

Last, for all the sentences that contained one of the 11 relational words of interest, the noun phrases recognized in the previous step that appeared before or after the relational words were extracted and put in relations using the relational words. The nltk package stored a dictionary of named entities, which were the nouns that were unique entities such as famous people’s names, city names, celestial bodies etc. All the unigram NPs extracted that were not unique entities will be excluded (e.g. saying ‘Dad is driver.’ is not of interest to this study although it might be fake news to some people). The extracted NPs were joint together by the relational words that denoted their relationship in the tweets to form (entity, relation, entity) triplets.

2.2) The researcher manually went through the extracted relations (triplets) to decide what relations actually satisfed the criteria specified for the current research (namely, relations that represented facts rather than opinions and that were representable with an NP-relation-NP triplet).

Facts, not Opinions. For the purpose of this study, the researcher only wanted factual statements, the statements in which people assert about the relationship between two objective entities. Factual statements are the things that people can talk “True” and “False” about. They should not be subjective opinions and should be ultimately verifiable. Unfortunately, the texts on twitter were mostly opinions. For example, the tweets “warm weather due to climate change is enjoyable, although it might kill us.” or “climate change is bad, but the weather is very very good.” were not necessarily stating any facts about climate change, but only expressing personal values. In addition, the statement could not be a commissive statement, a suggestion or a request, such as “we should stop global warming”.

What Kind of Facts to Extract. The current study only worked with simple, specific, literal, trait facts that are relatively demonstrably verifiable. The following were some examples of the kinds of facts that were not used for this study. Some facts were complex facts usually with multi-relationships between more than two entities and usually expressed with complex sentences. For example, “Biden said McConnell stopped Obama from calling out Russians” or “Trump stopped rule that wld have ended Reagan exception 2 law that native trees must be planted on former coal mines”. Some complex facts could be expressed in simple sentences as well. For example, “Idaho stopped teaching the link between fossil fuel and climate change.” or “Obama and Hillary sold Uranium to Russia”. But none of these simple sentences could be used for this study. These statements were statements of multiple facts on multiple levels. Some facts were general facts, assertions about relationship of general categories. For example, “Many politicians deny global warming…”. They were too vague and unfalsifiable. There was also considerable amount of sarcasm on twitter. For example, “#SmartNews : OH BUT GLOBAL WARMING IS FAKE NEWS! RIGHT?” or “The Earth is flat, Global warming is a lie and YouTube revenue is doing fine.”. Sarcasm is one of the things that are hard to give a strict definition to so that people can make the machine detect it. Sarcasm was excluded because people didn’t mean to assert the fact that they were literally expressing. Some facts were time/place/issue-specific facts. For example, “Republicans agreed with Democrats yesterday on bill XXX”. The truth values of such facts depended on the time and the specific events. They were excluded for the ease of verification. Last, some facts were metaphors such as “Fake news is a disease”. It’s hard to say it was false because it was not meant to be taken literally. All these kinds of facts were to be screened out by human coders after extraction and were to be excluded. Ideally, although not achieved in this research, researchers can also to a large extent automate 2.2 to better extract relations of desire. This is in itself another machine learning problem. For the current study, the selection was done completely manually.

3. **Learning and Prediction**

3.1) The dataset was split into two different parts, with the data from February as the training set and the data from March the testing set.

3.2) The truthfulness of the statements extracted in step 2.3 were verified for both training set and testing set. Each factual statement was verified by referencing snopes.com, factcheck.org, Wikipedia and google scholar. When the sources disagreed, a simple majority vote was used to decide the truthfulness of the statement. Mixed facts were treated as false.

3.3) A multilayer network was constructed for both the training set and testing set. For each of the 11 relational words, a network was built with the (entity, relational word, entity) triplets. All the entities that were extracted were used for all 11 networks, so the 11 networks had the same row names and column names in their adjacency matrixes. In each of the 11 networks, the entities’ true relation in terms of the relational word was constructed as a link. For example, in the network that was about the “cause” relationship, if one entity was stated to cause another entity and this statement was verified to be true, then a link was put between the two entities. Only true factual statements were represented with links. False statements did not correspond to any link in the network. After building the 11 networks, their adjacency matrixes were bound together into a three-dimensional multilayer matrix (a 3-d array), or, referred to as a tensor by Nickel.

3.4) Machine learning with tensor factorization introduced in I.3 were done on the training set. Let **X** denote the tensor representing the true relationship between entities on multiple layers (each layer is of one relational word). The statistical relational learning algorithm proposed by Nickel, Jiang and Tresp (2014) was used to do machine learning on **X** in the current study. After factorizing **X**, 3.5) the model parameters learned from the training set (matrix A, **R** and W) were used to make predictions about the testing set and create the model predicted values $\widehat{\boldsymbol{X}}_{test}$ according to equation 1:

$$\widehat{\boldsymbol{X}}_{test} = \boldsymbol{R} \times_1 A_{test} \times_2 A_{test} + \boldsymbol{M}_{\boldsymbol{test}} \times_3 W$$

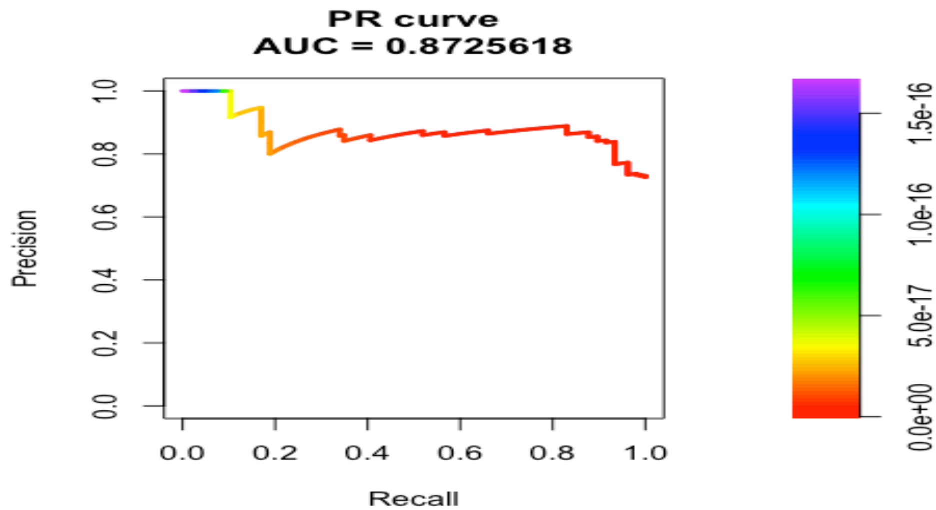
It was assumed that the same entity would have the same latent features in the testing set as in the training set. So, if an entity in the testing set existed in the training set, the values of latent feature vector learned from the training set was directly passed to the latent feature vector in the testing set. If a new entity appeared in the testing set, then the category of the entity (e.g. person's name, institution name, disease name, natural phenomena, place name etc) was considered. The average latent feature vector of the entities that fell into this category in the training set was assigned to the entity in the testing set. If the category of the entity was new too, the grand average of the latent feature vectors in the training set was given to the entity in the testing set. The W was exactly the same W for both training set and testing set. The **R** was exactly the same for both training set and testing set. After constructing A for the testing set using A from the training set, A, **R** and W were used to produce a predicted **X** for the testing set.

The intuition behind the prediction was that, for example, if the model learned that for entities to have a link on relation1, they usually need to both have high latent feature 2, entityA and entityB both had low latent feature 2, yet someone tweeted that entityA and entityB had a relaion1, then it can be predicted that it was highly likely to be a piece of fake news. In addition, if the model learned entities that had indirect relationship R1R2 were very unlikely to have relationship R3, and (entityA, R1, entityB) and (entityB, R2, entityC) were both true, then if in someone's tweet he or she said "entityA R3 entityC", it would very likely be fake news.

So after getting the model predicted values, the predicted values in $\widehat{\boldsymbol{X}}_{test}$ were compared to the observed values in **X** for the testing set. Because predicted **X** contained continuous values while observed **X** contains binary values (0 and 1), the way of comparing could not be a sum of squared errors. For this reason, an AUC test was used for an average performance of how well predicted **X** approximates the observed **X**. In the current study, an AUC-PR test was applied. AUC-PR tests have been shown to fit situations in which the proportion of positive cases was small.

4. Result

The result of the AUC-PR test showed that the automated detection was highly congruent with human coding. The AUC-PR test showed that the average prediction accuracy was 87%. This meant that the expected true positive rate of the algorithm's prediction was better than 50%, the baseline random predictor.



*What's learned in **R***. It's not clear what each dimension of A meant literally. For the sake of discussion, they will be referred to as latent feature 1 (LF1), latent feature 2 (LF2), and latent feature 3 (LF3). Generally, for relations of 1."agree", 2."disagree", 3."cause", 4."is", 5."faked/manipulated", 6."ruin", 7."deny", 8."kill", 9."reduce", 10."damage", and 11."is/are in", several patterns could be observed qualitatively from the estimated **R**. First, on relations 1."agree", 7."deny" and 11."is/are in", no particular pair of latent features made the two entities more likely to have the relation than others. This suggested that entities high or low in any of the three latent features could truly agree with, deny, or be in entities that were high or low in any of the three latent features. Second, on relations of 2."disagree"and 10."damage", entity pairs that were both high in LF1 were more likely to have these relations; on relation 3."cause", as long as the predicate (second entity) was high on LF1, the relation was more likely to exist as true. It could be conjectured that LF1 was a negative feature. This was also somewhat confirmed the layer of relation 6."ruin", where as long as the predicate was low in LF1, the relation was more likely to be true, because it's usually good positive things that are said to be 'ruined', and bad negative things usually said to be "caused". On layer 8."kill", any latent feature is likely to have the relationship with LF1. This could mean that the model learned that an entity with LF 1 is more likely true to be killed. Interestingly, on relation 4."is", when both entities have high LF1, the relation was less likely to exist. But generally relation 4 was similar to relation 1, 10 and 11, where no pair of latent features are particularly predictive for two entities to have this relationship. On the layer of relation 9."reduce", it was observed that an entity with LF2 was less likely to reduce an entity with LF2. On the layer of relation 5."faked/manipulated", it was observed that entities with LF3 were less likely to fake or manipulate entities with any latent feature. LF3 might be a positive feature. It's possible to conjecture about the underlying latent grammar of truth speaking. But this is only a qualitative crude version that is humanly interpretable. The accurate predictions, i.e. how likely exactly does having each feature make the entities have each relationship, are given by the model.

What's learned in W. One advantage of the ARE model is that it also utilizes indirect relationships between the entities to predict direct relationships for a higher accuracy. In the current study, four indirect relationships were picked: 1. "is - agree", the relation that entity1 is an entity that's agree by entity2; 2. "stop - disagree", the relation that entity1 stopped an entity that's disagreed by entity2; 3. "cause - fake", one entity causing something that is faked/manipulated by another entity; and 4. "kill - agree", one entity kills something that is agreed by another entity. The prediction strength of these four indirect relationships on the 11 direct relations were captured by matrix W. It was observed from W that indirect relation 1. “is - agree” predicted 1.“agree” and 3.“cause” stronger than predicting other direct relations. If A was something that’s agreed by B, then A was more likely to be agreed by B. More interestingly, if A was something that’s agreed by B, then it’s more likely to be caused by B than ‘damaged, or killed or is in” by B. Such patterns grammatically and literally made sense. Indirect relationship 2. “stop - disagree” predicted direct relationships 1,2,5,7,9,10,11 stronger than 3,4,6,8. It meant if A stopped things that were disagreed by B, then A was more likely to ‘be agreed”, “disagreed”, “faked/manipulated”, “denied”, “reduced”, or, “is/are in” by B than “caused”, “is”, “killed” or “ruined” by B. This again made sense literally. One might think B should thank A for stopping something that B disagrees, but “thanked” was not one of the relational words. When only considering the relational words at hand, compared to “kill” or “ruin” A, B was more likely to “reduce” or “disagree” with A. Similar logic applied to the other two indirect relationships. Indirect relationship 3. “cause - fake”, predicted direct relationship 2. “disagree” stronger than predicting other direct relationships. Indirect relationship 4. “kill – agree” predicted direct relationship 1, “agree”, 8, “kill” and 10. “damage” stronger than predicting other direct relationships.

Predictions on New Cases. There’s about 60 percent of the entities that appeared in the testing set but didn’t appear in the training set. The model achieved an average 87% prediction accuracy including these new predictions. Generally speaking, the model did well in predicting the new cases using the latent “grammar” learned in the training set.

5. Discussion

The machine learning method did not serve as an objective judgment for what counts as fake news, but more as a tool to automatically classify news according to people’s judgment patterns that people already formed but were sometimes hard to explicitly make into coding schemes. The algorithm could not judge what was fake for us by itself. We had to give it/show it our understandings of fakeness using examples in the training set and it would only give us what was consistent with our judgments in the training sample. The good part was that we didn’t need to explicitly tell the machine our rules of judgment. We only needed to give the machine a representative sample of well coded news pieces for it to learn our ways of judgments on its own.

Different from many other machine learning methods whose model parameters were not humanly interpretable, the ARE method provided results that were somewhat humanly interpretable. The patterns learned in A, **R** and W could be made sense of as some sort of latent grammar in the text corpus. And these “grammar” rules proved to predict the truth value of tweets better than chance.

The study included very many user-specified parameters and numbers. The extent to which how different choices of these numbers could have affected the result is not directly clear. The study applied a high degree of iterative interactions between human coding and machine computation, which required human to take the computer’s output as input, process it, and feed human’s output to the computer as input. The process could be time consuming and tedious. Future research should try to reduce the amount of human interpretation and make the process more automated.

One other limitation of the application is that it equates fake news with a non-existent link. However, a true relation that is not mentioned is also coded as a non-existent link. This method does no distinguish between the two. A more thorough listing of the facts as well as better selection of indirect relationships can be used to address these issues and improve the prediction accuracy.

III. Application to Congress Votes' Pattern Detection

1. Data

The voting behaviors of 1250 congress members on 7530 bills in the past 20 years, from congress 105 through congress 115 have been obtained from Congress' website using the ProPublica Congress API. The dataset contained much useful information, including information of the bills (e.g. type, topic, category), sponsors of bills, each member's vote on each bill, the voting results, members' information (such as partisanship, social media accounts) and so on. Some information was not retrievable from the Congress website and the obtained dataset was only a subset of all votes happened in the past 20 years. Two networks were built from this dataset, a co-sponsor network and a co-vote-for network. Both networks had all the congress members as nodes. The co-sponsor network put an edge between two nodes whenever the two nodes sponsored for the same bill. The weight of each edge was the number of times the two members co-sponsored in the dataset. The co-vote-for network put an edge between nodes that both voted for same bills. The weights of the edges were the number of bills the two nodes both voted for.

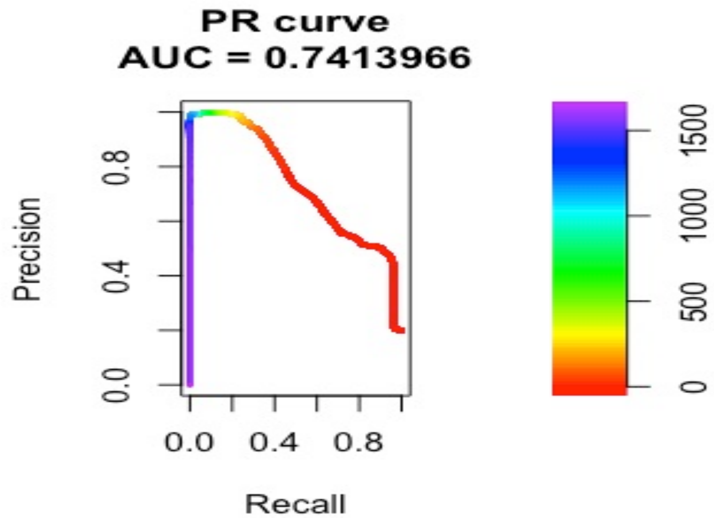
Based on the information in the obtained dataset, the tweets of the congress members (those who use twitter) were retrieved from their twitter accounts. A "twitter interaction" network was built with the twitter data. Each node in this network was a congress member and a link was established between two nodes if one the two members ever replied to the other's tweets. The links were weighted, with the weight being the number of times the pair has had interaction on twitter. The twitter interaction network was sparse. On average, one congress member had twitter interaction with about 1.4 other congress members.

2. Analyses

The 7530 bills were randomly split into two parts, one part with 5269 (70%) bills as the training and one part with 2258 (30%) bills as the testing set. The members' voting pattern on the 5269 bills were learned by the model and applied to make predictions on the testing set. In this case, tensor **X** was made up of three layers, the co-sponsor layer, the co-vote-for layer, and the twitter interaction layer. Each layer was a one mode network with the congress members as row and column names. Matrix A was again specified as the latent features of each congress member, each with 3 dimensions. Three indirect relationships of members were chosen, 1. "sponsor-for", 2. "sponsor-tweet", 3. "for-tweet", respectively representing relationships that a member co-sponsored with someone that another member co-voted-for with, a member cosponsored with someone that another member interacted on twitter, a member co-voted-for with someone that another member interacted on twitter with. How the latent features and indirect relationships predicted the values of links in **X** was learned in the training set and applied to the testing set for validation.

3. Result

An average true positive rate of 74% was achieved in the testing set.



In the testing set, only one congress member was new (not in the training set). That member's cosponsors' average latent feature vector was given to him/her. The bills in the testing set were all new. The model learned the voting pattern of the same set of members on one set of bills and applied it to predict their voting behavior on a totally new different set of bills. The prediction proved to work better than chance.

A and R. Matrix A contained each member's values on three latent features that are predictive of their relationships. Tensor **R** contained how much each pair of latent features, if present, predicts each of the three specified relationship. The values in tensor **R** indicated that members that were both high in latent feature 3 were more likely to have a cosponsoring relationship. Combinations of latent feature 1 and 1, 2 and 2 or 1 and 2 predicted co-vote-for relationship better than combination of 1 and 3 or 2 and 3. So the two members need to be high in either latent feature 1 or 2 but not 3 in order for them to vote for bills together. On the third layer of R, latent feature combinations of 1 and 1, 2 and 2 or 3 and 3 tended to predict twitter interaction better than combinations of 1 and 2, 1 and 3 or 2 and 3, indicating that two members needed to be high in the same latent feature categories to be more likely to interact on Twitter.

Matrix W. Matrix W contained the strength of relation between each of the indirect relationships (1. Sponsor + for, 2. Sponsor + twitter interact, 3. For + twitter interact) and each of the direct relationships to be predicted (1. Co-sponsor, 2. Co-vote-for, 3. Twitter interaction). Matrix W showed that indirect relationship 1. "cosponsor + for" predicted direct relationship 2. Co-vote-for stronger than other relationships did, meaning that if member A sponsored with someone that agreed with member B on a bill together, then A and B are also more likely to vote for bills together than cosponsor or interact on twitter. Indirect relationship 2. "cosponsor + twitter interaction" had weak relationships with all direct relations but relatively speaking, it had a stronger relation with direct relation 3. "twitter interaction" than the other two. Indirect relationship 3. "for + twitter interaction" strongly related to direct relation 2. "co-vote-for" more than to other direct relationships. If member A agreed with someone on bills who had twitter interaction with member B, then member A is more likely to agree with member B on bills than co-sponsor or have twitter interaction with B. Among all the three indirect relationships, 1. "cosponsor + for" was the strongest predictor for 1. "cosponsor"; indirect relationship 3. "For+twitter interaction" was the strongest predictor for 2. "co-vote-for" and 3. "twitter interaction" behaviors.

4. Discussion

Even without using exogenous information such as members' partisanship, or the bills' category and content, but only based on the voting behavior itself, the voting behavior turned out to be predictable. The result showed that congress members' votes were a consistent behavior and the pattern of voting could be picked up by machine learning methods. In the current study, all the votes were aggregated and randomly split into training sets and testing sets. It might be interesting to further explore if the pattern has changed over time.

IV. General Discussion

1. Regression Appeals to Population Level Parameters while Relational Machine Learning Appeals to Individual Traits

The two applications of the machine learning algorithm achieved acceptable level of accuracy. Although same level of prediction accuracy may be achieved with traditional linear regression analyses, the machine learning method still makes a useful alternative approach. What is unique about the machine learning algorithm is that its predictions are point-wise. Simple linear regression or logistic regression only could provide an estimate for the population level parameters and apply the estimated population level parameters onto each individual to make predictions. But it could not explain why any given specific individual case took the value it took except for pointing out the fact that that case was part of the population and that value was what's expected for that case in that population. In other words, the linear regression assumes that all cases shared the same set of parameters (regression coefficients). If two cases had the same predictor values, they would have the same predicted values. The machine learning method, on the other hand, estimates the individual parameters for each individual case (e.g. each link) and allows each case to be predicted locally. For example, two pairs of members could have same number of co-sponsoring times in the training set but still have different model predicted number of co-sponsoring times. This method gives more insights into why a certain pair had the relation they had by appealing to local individual information of the two individuals in that pair rather than by appealing to an average general trend observed in the whole population.

2. Machine Learning Results Are Not as Humanly Interpretable as Regression

In many neuro-network algorithms, the researcher won't know how exactly the machine interprets the rules or patterns learned from the training set and how exactly what's learned in the model parameters could explain the predictions based on them. Simple linear regression on the other hand, due to its assumptions about the direct linear relationship between the input and output variables, provides easy-to-interpret coefficients. The current algorithm lies in between the two. With the current algorithm, researchers can know how the machine did the learning and made the prediction because the structure of what to learn was user specified (i.e. latent features and indirect relationships), By telling the machine to factorize the training set into products and sums of latent features and indirect paths, the learning results and prediction could be largely made sense of. However, the accurate meaning of each latent feature is still hard to pin down and put into human language.

3. Possible Pitfalls

The 'garbage in, garbage out' metaphor still applies to even the state-of-the-art algorithms. If the data were biased due to noise, data obtaining errors or other factors, the conclusions may be biased or completely wrong. When dealing with the unstructured data of human language, the data cleaning process including human coding could introduce unrealized artifacts and their effects are unknown. In addition, many parameters of the method were user specified. It still remains an empirical question how these arbitrary parameters affect the result of this or any future study. One definite improvement of the study would be to better obtain, clean and structure the data and use better selection of parameters.

V. References

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Cho, H., & Lee, J. S. (2008). Collaborative information seeking in intercultural computer-mediated communication groups: Testing the influence of social context using social network analysis. Communication Research, 35(4), 548-573.

Nickel, M., Jiang, X., & Tresp, V. (2014). Reducing the rank in relational factorization models by including observable patterns. In *Advances in Neural Information Processing Systems* (pp. 1179-1187).

Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. Proceedings of the IEEE, 104(1), 11-33.

Rubin, V., Conroy, N., Chen, Y., & Cornwell, S. (2016). Fake news or truth? Using satirical cues to detect potentially misleading news. In Proceedings of the Second Workshop on Computational Approaches to Deception Detection (pp. 7-17).

Robins, G., Lewis, J. M., & Wang, P. (2012). Statistical network analysis for analyzing policy networks. Policy Studies Journal, 40(3), 375-401.

VI. Appendix

R (fake news)

```
, , 1
              [,1]          [,2]          [,3]
[1,] -1.464844e-132 -1.279759e-132 -3.236845e-132
[2,] -2.756513e-132 -2.408191e-132 -6.090951e-132
[3,] -1.993223e-132 -1.741301e-132 -4.404220e-132

, , 2
              [,1]          [,2]          [,3]
[1,] -5.392291e-130 -8.639022e-130 -5.574965e-130
[2,] -1.211934e-129 -2.000924e-129 -1.322699e-129
[3,] -1.000888e-129 -1.683799e-129 -1.129189e-129

, , 3
              [,1]          [,2]          [,3]
[1,]  2.189996e-131  4.115370e-131  2.972331e-131
[2,] -5.595957e-131 -1.051573e-130 -7.594991e-131
[3,] -1.139442e-130 -2.141201e-130 -1.546484e-130

, , 4
              [,1]         [,2]         [,3]
[1,] -6.065254e-141 7.830727e-140 1.398469e-139
[2,]  1.980994e-139 6.213250e-139 6.697022e-139
[3,]  2.987620e-139 8.080192e-139 7.981473e-139

, , 5
              [,1]          [,2]          [,3]
```

```
[1,] -1.162822e-133 -6.456212e-133 -1.540850e-132
[2,] -2.185758e-133 -1.214662e-132 -2.899650e-132
[3,] -1.579059e-133 -8.782779e-133 -2.097137e-132

, , 6
               [,1]           [,2]           [,3]
[1,] -1.160451e-133 -2.206073e-133 -1.611493e-133
[2,]  3.847386e-133  7.326116e-133  5.354751e-133
[3,]  7.605445e-133  1.447210e-132  1.057336e-132

, , 7
              [,1]          [,2]          [,3]
[1,] 2.619430e-132 1.884438e-132 4.296293e-132
[2,] 4.923226e-132 3.541813e-132 8.074906e-132
[3,] 3.556393e-132 2.558505e-132 5.833081e-132

, , 8
               [,1]           [,2]           [,3]
[1,]  2.865000e-134  5.436776e-134  3.964118e-134
[2,] -7.973680e-133 -1.509120e-132 -1.097540e-132
[3,] -1.499923e-132 -2.838779e-132 -2.064551e-132

, , 9
               [,1]           [,2]           [,3]
[1,] -5.936269e-135 -8.904486e-135 -5.185958e-135
[2,] -8.795084e-135 -1.319369e-134 -7.747882e-135
[3,] -5.028170e-135 -7.543636e-135 -4.482691e-135

, , 10
               [,1]           [,2]           [,3]
[1,] -3.926843e-137 -2.197953e-136 -5.251878e-136
[2,] -6.480298e-137 -3.620851e-136 -8.656777e-136
[3,] -3.159130e-137 -1.801197e-136 -4.317370e-136

, , 11
              [,1]          [,2]          [,3]
[1,] 1.502064e-139 3.434819e-139 3.077528e-139
[2,] 4.255131e-139 9.606739e-139 8.564865e-139
[3,] 4.283454e-139 9.642453e-139 8.586556e-139
```

W (fake news)

	V1	V2	V3	V4
1	1.993692e-09	3.024548e-09	3.811174e-09	2.836224e-09
2	4.051613e-10	2.432220e-09	3.431475e-08	5.110869e-10
3	1.560792e-09	9.299271e-10	4.474417e-15	4.778760e-10
4	5.934392e-10	3.385733e-10	3.430493e-09	5.935752e-10

5	5.809054e-10	3.435827e-09	3.892015e-09	8.704880e-10
6	6.787555e-10	2.821514e-10	3.928915e-15	6.022204e-10
7	4.615978e-10	1.554170e-09	7.503082e-09	2.977119e-10
8	4.567732e-10	3.394074e-15	4.067535e-15	2.815715e-09
9	4.803427e-10	8.596003e-09	1.804022e-09	3.257294e-10
10	7.947273e-10	9.540157e-09	9.868841e-10	1.029410e-09
11	7.876262e-10	1.099604e-09	7.157030e-10	5.470617e-10

R (congress votes)

, , 1

	[,1]	[,2]	[,3]
[1,]	-0.00533837	-0.03831606	-0.04724963
[2,]	-0.07583233	-0.05530078	-0.00819963
[3,]	-0.06270600	-0.01080590	0.04493186

, , 2

	[,1]	[,2]	[,3]
[1,]	327.72774	293.9704	12.89082
[2,]	291.26675	487.2839	171.26309
[3,]	11.93784	171.3492	183.28852

, , 3

	[,1]	[,2]	[,3]
[1,]	0.0005031278	7.671341e-05	-0.0006172721
[2,]	0.0001237007	4.694627e-04	-0.0004814778
[3,]	-0.0001841670	1.032954e-04	0.0002572132

W (congress vote):

	spsfor	spstwt	fortwt
cosps	5.907177e-06	-3.23408e-02	4.357925e-06
cofor	4.483447e-04	-3.63471e+00	1.918579e-03
cotwt	-1.07487e-09	-8.89567e-05	3.601282e-06