Project Proposal

Data Science 670

Adam Noack

March 22, 2019

1 Introduction

Nodes in a graph can represent entities in a network, and the relationships between these entities can be represented by edges between entities. Classical Social Network Analysis (SNA) allows one to analyze the structure and content of networks represented as graphs. It gives one the ability to determine the effect that the interconnections of a network have on each entity and provides information about how the network as a whole and its individual components may evolve over time.

Nodes can represent more than just people, though. Edges, too, can represent a wide variety of interesting things. Indeed, SNA can be applied in many different contexts. In this day and age, ever more things in our lives are transmitting and recording data and connecting with each other. Whether it is profiles on an online social network, sites on the web, or our refrigerator and TVs in our home, the web of interconnection is growing and strengthening. Thus SNA offers us many opportunities to better understand the world around us.

2 Applications of SNA

2.1 Application of Social Network Analysis to Collaborative Team Formation

In this paper, the authors attempt to use SNA to find groups of scientists that are working on similar problems so that each scientist knows which other scientists might make good collaborators [2]. They use a dataset consisting of 71 papers written by 80 different authors from the Air Force Research Laboratory between 2003 and 2005.

First, the authors use traditional SNA to build a graph in which each node is an author and every edge indicates the two authors it connects collaborated on a paper at some point. When viewing the graph in two dimensions, it is immediately clear that there is a high degree of clustering. I.e. there is often collaboration within, but not between, groupings of individuals.

The authors then analyze the papers that were written. They assign each paper a few keywords that encapsulate its general theme. They then create a concept map of these various keywords and lay it on top of the graph of individuals. In doing this, they find that there are individuals working on papers with similar concepts across groups of individuals. Using the insights gleaned from their analysis, these individuals were more likely to connect and avoid duplication of work.

2.2 Website Clustering from Query Graph using Social Network Analysis

In this paper, the authors seek to cluster websites [7]. Clustering of websites makes the task of avoiding or finding certain websites much easier. Of course, this is not the first attempt to cluster websites, however, their particular approach to clustering

is unprecedented. Typically clustering of websites is done by having the edges in the graph represent hyperlinks from one website to another. Wang et al. build a graph of websites in a database by classifying each website by the search query that returned it. They do this in the following manner: they start with a database consisting of 21 million query / returned url pairs. They then take each query and represent it as a 21 dimensional vector. Each dimension in the vector represents the amount of times words in the search query have fallen into that particular category. Some of these categories include: "economics", "education", "sports", "food". These vectors describing each query are then used to represent the url that each query returned. Then the cosine similarity measure between each of the 21-dimensional vectors is performed on every possible pairing of websites. If the cosine similarity is greater than a certain threshold, an edge is created between the websites in their graph.

The authors then use the LPA and fast GN algorithms to cluster the websites. With their algorithm websites are nicely clustered in such a way that it will be easy for users of the search engine to navigate to useful websites and avoid unproductive and even malicious urls.

2.3 Interconnectedness of Complex Systems of Internet of Things through Social Network Analysis for Disaster Management

In this paper [8], the authors notice that previous attempts to model the IoT as a linear hierarchy of sorts are incomplete. They propose that the devices that make up the IoT should be modeled as a complex, multi-dimensional system of interlinked objects. They believe that adopting a social network analysis approach when modeling the IoT will provide new insights about the structure and dynamics of the IoT.

In the context of disaster management, the authors argue that many of the devices used to communicate between individuals can be treated as edges in a network, and the nodes can represent individuals described by the output of various on-site digital technologies.

This paper does not produce anything new, rather, it describes a vision for the future – a future in which all of the devices plugged into the giant web that is the IoT can act and communicate in concert using techniques derived from SNA.

2.4 Trust Based Knowledge Outsourcing for Semantic Web Agents

Humans learn about the world by observing patterns in the structure of reality directly and also by querying people around them when unsure of something. I.e. people learn about the world from first-hand experience of the world, or second-hand, through the experiences of others. Of course, people only adjust their world view based on the ideas of others when there is a reasonable amount of evidence suggesting that the sources are trustworthy. On the semantic web, agents are stuck with the task of choosing which source of information to accept as truth. In a manner similar to the way we humans select which sources of information are to be trusted, agents on the semantic web must select sites that are most likely to be trustworthy.

In this paper [4], the authors create a simulated environment of agents where each agent begins with a domain knowledge base consisting of truth values for a set of propositions. At each timestep in the simulation, some agents request information about the environment. They choose which other agent to trust in the following manner. Agent A will trust agent B if A has had positive experiences in the past with B. If agent A has not yet had any encounters with agent B, then agent A

estimates his trust in B by randomly selecting a set of agents NA and finding the level of trust each agent $N \in NA$ has in B weighted by the level of trust A has in each N and averaging these values.

2.5 Supervised Learning of Universal Sentence Representations from Natural Language Inference Data

The goal of the authors in this paper [3] is to create a general-purpose sentence encoder that is applicable across a wide range of NLP tasks. To do this, they train various models in a supervised manner to generate embeddings for the sentences in the Stanford Natural Language Inference (SNLI) dataset. The authors hypothesize that the natural language inference task is best for creating general purpose embeddings as the task requires high level semantic information to be completed successfully. They found that the best performing embedder architecture was the bi-directional LSTM model, and the results obtained using this architecture eclipsed those of previous models such as the Skip-Thought model.

3 My project

3.1 Motivation

The number of humans active on online social networks is massive, and it is only growing. In 2017 there were 2.46 billion active users, and by 2021, it is predicted that there will be nearly 3.02 billion [1]. This means that people are spending ever more time connecting and communicating with friends and acquaintances on various social media platforms with their devices. People are effectively replacing conversations and interactions that have historically taken place face-to-face with digital dialogue [6].

Because of this, more and more of our social lives are being tracked and recorded. Much of the data that the users of networks such as Facebook, Twitter, and Instagram produce is publicly available. Insights drawn from these rich networks could prove highly valuable. Knowing how to aggregate the information present in in each user's ego-network to generate predictions for an individual user will be highly valuable. However, aggregating this data presents some challenges. It is often difficult to know which way is best to combine the feature values for various users. Also, in the real world, the process of aggregating the information present in each person's ego-network may be time-intensive and tricky because all of the feature values for each node in the ego-network must be kept in sync in some manner.

3.2 Dataset

I worked with online social network data for my project. Specifically, I used the Pokec graph dataset from Stanford [5]. It consists of anonymized Pokec profiles and each profile's list of friends. There are over 1.6 million nodes in the network, or profiles, and over 30.5 million edges, or "friend" connections, linking these nodes. Every user has information describing their age, registration date, last login time, and gender. A certain subset of the 1.6 million users also provided answers to certain various personal questions about their hobbies, favorite movies, profession, etc. I wished to use the answers to a few of these questions to predict the age for a user. I decided to use the answers to the following subset of questions to predict age: movie preference, hobbies, profession, children. I also included each individual's gender as a feature.

3.3 Data preprocessing

3.3.1 Whiddling down the dataset

I removed all of those users that did not answer all of the four questions. This left me with a dataset containing a little over 250,000 users. Next, I removed those users that had invalid age values and was left with a dataset of roughly 170,000 users.

3.3.2 Dealing with answers written in Slovak

All of the answers to the four questions were written in Slovak. I initially wanted to search for words that appeared often for each answer and then create new binary features indicating the presence or absence of these words. However, there didn't seem to be much of pattern to the responses. They were quite freeform: some were lists, others were in sentence form, and still others were lists of sentences. This fact, among others, led me to believe that embedding each response using a sentence embedder would result in the smallest amount of information loss. However, there was no pre-trained, high-performing Slovakian sentence embedder. So, I made use of the googletrans library from Google. My plan was to translate all of the aggregate responses for each of the 170,000 users into English and then use an English sentence embedder. However, in this latest version of the googletrans library, a translation request limit was placed on the API. Therefore I could not translate more than about 500 sentences per day. To combat this issue, I concatenated each of the four responses for each user together and then translated the resulting aggregate sentence. Still, I was only able to translate a little over 1,500 sentences in total. Thus my dataset was effectively reduced from 170,000 to 1,548 samples.

3.3.3 Using the InferSent sentence embedder

After translating the aggregate response sentences into English for the 1,548 users, I embedded each word in each sentence in 300 dimensional space using a GloVe word-to-vec model. Then I used the InferSent Bi-LSTM trained on the Stanford Natural Language Inference dataset [3] to embed each sentence in 4,096 dimensional space. Thus each user had 4,096 features plus one for their gender.

3.3.4 Aggregating the data for each ego-net

It was my hope that the responses that each user and his/her friends wrote would in some way be correlated with their age. Explicitly, my hope was that the feature values for one individual plus the feature values of that individual's friends would together provide more information than just the one individual's feature values. Thus I found the average sentence embedding for each user's friends and then appended this average embedding to the user's list of features. This resulted in a 8,194 dimensional vector for each user containing information about each user's own profile and that of his/her friends.

3.3.5 Conversion from regression to classification problem

Instead of treating the age prediction task as a regression one, I decided to bucket the ages. Users with an age less than 20 were given the label 0, users between the ages of 20 and 23 were given the label 1, and people older than 24 were given the label 2. 43% of the users fell into the middle age bracket, 31% in the youngest bracket, and 26% in the upper bracket.

3.4 Model selection and creation

The feature space for this problem was remarkably similar to those found in natural language inference tasks, so I used a neural network model architecture similar to that found in the InferSent paper. Specifically, I used Keras with a TensorFlow backend to create a standard feed-forward neural network with one hidden layer with 512 units followed by a dropout layer with a dropout probability of 0.6 culminating in a 3-way softmax, one for each age bucket.

3.5 Training details

My train, validation, test split proportions were .72, .12, and .15, respectively. The model utilized categorical cross entropy as a loss function and the Adam optimizer. The model was trained on the data that has the aggregate embeddings attached for nine epochs with a minibatch size of 32. The model was trained on the data that does not have the aggregate embeddings attached for four epochs with a minibatch size of 32.

3.6 Results

Data	Train accuracy	Test accuracy
HAS friends' aggregate embedding	60.5%	56.2%
LACKS friends' aggregate embedding	52.6%	50.2%

As can be seen, the model does a better job of predicting the age of the user when it has access to the features of the user's friends. However, given the distribution of labels in this dataset, 43% accuracy can be achieved if the model guesses the middle age bracket every time. Therefore, this model performance, while being better than informed guessing, leaves much to be desired.

3.7 Discussion

There are a number of things I would have liked to do that may have increased the accuracy of the model.

- Translate *many* more of the users' aggregate sentences from Slovak into English. Increasing the training set size usually increases the model accuracy.
- When aggregating over the embeddings for each user's friends, weight more highly the embeddings of those friends with which triangles between other individuals in the ego-network are present. I.e. attempt to weight the features of close friends more.
- Clean the translated sentences better before embedding.
- Use a different subset of the original features.
- Instead of using the mean aggregate function to combine the embeddings for each of the users, use max. Mean pooling might destroy the significance of some feature dimensions.
- Formulate the task as a regression problem instead of bucketing the age values and treating it as a classification task.

4 Conclusion

It is clear that social network analysis can be used in a variety of different ways to draw useful conclusions. As the number of connected devices on this planet continues to increase, and an increasing percentage of our lives are digitized and recorded, this method of analysis will become ever more valuable. As has been shown, the

information describing our friends has much to say about ourselves. The old adage that one is the average of those people with which he spends the most time with is quite true, and if companies and governments of the future truly want to know their people well, it would be in their best interest to get to know each person's friends too.

References

[1] Mobile social media - statistics and facts. *Statista*.

[2] Michelle Cheatham and Kevin Cleereman. Application of social network analysis to collaborative team formation. In *Proceedings of the International Symposium on Collaborative Technologies and Systems*, CTS '06, pages 306–311, Washington, DC, USA, 2006. IEEE Computer Society.

[3] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364, 2017.

[4] Li Ding, Lina Zhou, and Tim Finin. Trust based knowledge outsourcing for semantic web agents. pages 379– 387, 11 2003.

[5] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[6] Kalpathy Subramanian. Influence of social media in interpersonal communication. *INTERNATIONAL JOURNAL OF SCIENTIFIC PROGRESS AND RESEARCH (IJSPR)*, 109:pages 70–75, 08 2017.

[7] Quoc-Dinh Truong, Taoufiq Dkaki, and Quoc-Bao Truong. Graph methods for social network analysis. volume 168, pages 276–286, 03 2016.

[8] Asta Zelenkauskaite, Nik Bessis, Stelios Sotiriadis, and Eleana Asimakopoulou. Interconnectedness of complex systems of internet of things through social network analysis for disaster management. pages 503–508, 09 2012.