Joint Embedding Models for Textual and Social Analysis

Chang Li [1] Yi-Yu Lai [1] Jennifer Neville [1] Dan Goldwasser [1]

Abstract

In online social networks, users openly interact, share content, and endorse each other. Although the data is interconnected, previous research has primarily focused on modeling the social network behavior separately from the textual content. Here we model the data in a holistic way, taking into account connections between social behavior and content. Specifically, we define multiple decision tasks over the relationships between users and the content generated by them. We show, on a real world dataset, that a learning a joint embedding (over user characteristics and language) and using joint prediction (based on intra- and inter-task constraints) produces consistent gains over (1) learning specialized embeddings, and (2) predicting locally w.r.t. a single task, with or without constraints.

1. Introduction

The remarkable popularity of social media outlets provides exciting opportunities to study language and social behavior on a large scale. These outlets allow users to openly interact, share content, endorse and disapprove of the behavior and stances of each other. The interconnected structure of this data strongly suggests that it should be studied in a holistic way, taking into account the connections between the social behavior of users, their stances and viewpoints, and the content generated when they interact with other users. Taking this approach would assist social network researchers to understand the communication patterns between users in the network. In addition, it would also allow us to study natural language in a holistic way, taking into account the social context in which it was generated.

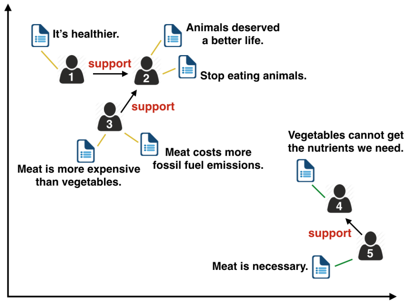
It's healthier.
Animals deserved a better life.
support
1
2
Stop eating animals.
3
support
Meat is more expensive than vegetables.
Meat costs more fossil fuel emissions.
Vegetables cannot get the nutrients we need.
4
support
5
Meat is necessary.

Figure 1. Illustration for proposed model on Debate dataset.

In this paper we suggest a holistic approach, combining users' information, their social behavior and language use, into a joint model. Unlike previous works, which model the connections between these aspects either by using collective classifications approaches *or* by creating user embeddings based on their social structure, in this work we suggest that the two can be combined, and show empirically the advantage of this combination.

The broadly applicable settings of our model can be represented as a graph (illustrated in Figure 1), connecting users with one another using edges that capture social behaviors, and users with text, which is reflective of their opinions, age (and other attributes) and social goals. Our model uses this graph in two ways. (1) We embed users and their textual content in a single vector space (Section 2) based on the graph structure. This representation utilizes the social signal represented in the graph edges. In addition, we extend previous work (Tang et al., 2015), by also incorporating different types of social signals between users, such as the text of messages they post, and the attributes they share. (2) We define multiple decision tasks over the relationships between the users and the content generated by them (section 3). We define these predictions over the embedding space, by comparing the similarities of pairs of vectors representing objects. We exploit the graph structure to perform collective classification to ensure consistency between these decisions. For example, in Figure 1, *user 1* supports *user 2*. Identifying this relationship can inform us about the relationship between *user 1* and the content generated by *user 2*.

Related Work Several recent works have explored embedding methods for social network analysis (Perozzi et al., 2014; Grover & Leskovec, 2016; Tang et al., 2015), by learning the network embedding through exploring node

[1] Purdue University. Correspondence to: Dan Goldwasser <dgoldwas@purdue.edu>, Jennifer Neville <neville@purdue.edu>.

Proceedings of the 34^th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

neighborhoods (contexts). Several works utilize social information for NLP problems. For example, (Benton et al., 2016) combines multiple views into a single embedding, in (Yang & Eisenstein, 2015) community-specific projections of word embedding are used. There is little work looking into a shared model connecting text, attributes and behavior. The closest to our work is (Li et al., 2015), that jointly integrates different kinds of cues (text, attribute, graph) into a single latent representation.

2. Learning

Our first step towards jointly modeling social interaction and language is to learn a joint embedding of user characteristics and language. Our embedding method is similar in spirit to the Deep Structured Semantic Model (DSSM) (Huang et al., 2013).

Let U denote the set of all users, let A and T denote the set of all attribute vectors and text authored by those users respectively. Our objective is to learn a semantic embedding for both users text and user attributes so that they are close in the embedding space if they are semantically close to each other. For a text (or attribute) input x, we will compute its embedding e using M hidden layers $l_i, i = [0, M-1]$.

To learn the embedding, we will divide a large social network into small sub-networks. All training and test examples are generated from within each sub-network. In our debate dataset, it is natural to consider a debate between two participants as a sub-network. Hence, we have the two users (which we refer to as **author**), arguments from multiple rounds in the debate (which we refer to as **text**), and other users who voted for either side (which we refer to as **endorser**) in the sub-network.

2.1. Embedding Views

Given the scenario described above, we can now describe several different objectives that we can consider while learning the embedding. Each of them focuses on two contending relations (edges in the graph).

Text vs. Attribute (TA): This objective is to distinguish text t_i written by an author u_i from text t_j that is written by another user $u_{j \neq i}$, by using the attribute representation of u_i (i.e., based on a_i). For the debate dataset, positive examples consist of (a_i, t_i) pairs (attributes and text from one round of the debate of user u_i). Negative examples consist of (a_i, t_j) pairs, where t_j is the text in the same round but from the contender in the debate.

Text vs. Text (TT): This objective is to distinguish text t_i^a written by an author u_i from text t_j that is written by another user $u_{j \neq i}$, by using the language representation of u_i ($t_i^{b \neq a}$). We consider adjacent debate rounds or posts (e.g., t_i^b, t_i^a) as positive examples for this objective. For negative examples, we use text from the contender (e.g., t_i^b, t_j).

Attribute vs. Attribute (AA): This objective is to distinguish author u_j supported by endorser u_i from the author u_k that is not, using the attribute representation of each user (e.g., a_i). Positive examples consist of (a_i, a_j) pairs where u_j voted for u_i in a debate. Negative examples consist of (a_i, a_k) pairs, where u_k voted against u_j.

2.2. Embedding Objective

Given two pairs of objects, one pair o, c^p corresponding to a positive example and one pair o, c^n corresponding to a negative example (as defined by the embedding views), we can define our embedding loss for each view –

$$L_V = \sum_{(o_i, c_i^p, c_j^n)} l(sim(e_{o_i}, e_{c_i^p}), sim(e_{o_i}, e_{c_j^n})) \quad (1)$$

In the Text vs. Attribute view, o corresponds to a user's attribute vector (a), and c^p, c^n correspond to texts t^p, t^n written by the user and by a different user, respectively.

We consider the similarity among the embeddings e_{o_i}, $e_{c_i^p}$, and $e_{c_j^n}$. The goal is to maximize the similarity between the user's attribute and textual representation, while minimizing the similarity between the user's attribute representation with the textual representation of another user. $sim()$ is a vector similarity function; we use cosine in this work. $l()$ is the cross-entropy loss.

$$l(p, n) = -\log\left(\frac{e^p}{e^p + e^n}\right)$$

Joint Embedding Loss Function: We combine the embedding objective for each view into a joint training objective:

$$L(U, T, A) = L_{TA} + \lambda_1 L_{TT} + \lambda_2 L_{AA} \quad (2)$$

Similar to the Structure-preserving constraints proposed in (Wang et al., 2015), we also take advantage of these implicit constraints in our joint embedding model.

3. Prediction

Our joint embedding model maps users (represented as attributes vectors) and text into the same space. This mapping reflects the users attributes, the relationships between them and the content they authored, allowing us to compute the similarity between any pair of users, texts or their combination. This is a useful property, as multiple prediction tasks can be defined over this similarity metric without requiring further retraining, simply by defining these tasks as "multiple choice" predictions which can be decided by comparing the similarity scores of candidate pairs.

In this paper we consider four such tasks, capturing relationships between users and text. These prediction tasks are highly inter-dependent. For example, if we know two texts are written by the same author, then identifying the author of the first piece of text also determines the authorship of the other one. We exploit these dependencies by making a joint prediction over multiple instances. We formulate the decision as an Integer Linear Program (ILP) which allows us to directly force the consistency between decisions.

3.1. Prediction Tasks

We define multiple prediction tasks, each requiring the model to decide between two alternatives. In all cases, one of the candidates will result in a correct decision.

AuthoredBy Given a user u, and two text candidates t_p and t_n, predict which one is authored by u.

SameAuthor Given three text candidates t_i, t_j and t_k, predict which of the latter two is from the same user as t_i.

AgreeWith Given three users u_i, u_j and u_k, predict which of the latter two u_i agrees with.

SupportedBy Given one user u_v, and two texts t_p and t_n, predict which text u_v supports.

Similar in spirit to (Amid & Ukkonen, 2015), we define an instance of the above four tasks as the triad (x_i, x_j, x_k) of items where x_i is called the probe item and x_j and x_k are called the test items. We can make the decision locally by comparing $sim(e_{x_i}, e_{x_j})$ and $sim(e_{x_i}, e_{x_k})$, and predicting the relation holds for the semantically closer pair.

3.2. Joint Prediction

We model the dependencies between decisions by predicting them jointly. To do so, we will define the predictions over the sub-network of any two given users as an Integer Linear Programming (ILP) instance.

The ILP global optimization is defined over two given users u_i, u_j, the textual content generated by them respectively, $\{t_i^0, ..., t_i^k\}$, $\{t_j^0, ..., t_j^k\}$, and other users $\{u_i^0, ..., u_i^m\}$, $\{u_j^0, ..., u_j^m\}$, who have *supported* u_i, u_j respectively.

We create four types of boolean decision variables corresponding to the tasks above. Specifically, we associate a boolean variable $\alpha_{k,l}$ with each one of the users ($k = \{i, j\}$), and the text t_k^l, and associate a score $sim(e_{u_k}, e_{t_k^l})$ with that variable. Similarly, we associate a boolean variable $\beta_{i,j}$ with every two texts, and associate a score $sim(e_{t_i}, e_{t_j})$ with it. Another boolean variable $\gamma_{k,l}$, for any two users u_k, u_l, who have *supported* either u_i or u_j, and associate a score $sim(e_{u_k}, e_{u_l})$ with it. Finally, we associate a boolean variable $\delta_{k,l}$, with pairs consisting of a user u_k, who has *supported* either u_i or, u_j, and text t_l, and associate a score $sim(e_{u_k}, e_{t_l})$ with it. The set of all possible decisions for the tasks are denoted as A for the AuthoredBy task, B for the Same Author task, Γ for the AgreeWith task and Δ for the SupportedBy task.

Given these variables, our prediction function is:

$$\arg\max_{\alpha,\beta,\gamma,\delta} \sum_{\alpha \in A} \alpha \cdot score(\alpha) + \sum_{\beta \in B} \beta \cdot score(\beta)$$
$$\sum_{\gamma \in \Gamma} \gamma \cdot score(\gamma) \sum_{\delta \in \Delta} \delta \cdot score(\delta)$$

Subject To **C**

Where **C** is a set of constraints defined as follows:

Intra-task constraints: We define two types of intra-task constraints that restrict the decisions within one task.
(1) Given two users u_1, u_2 and text t from one of them:

$$\texttt{AuthoredBy}(t_1,\ u_1) + \texttt{AuthoredBy}(t_1,\ u_2) = 1$$

(2) Given a pair of texts with different authors, t_2 and t_3, and another text t_1 sharing an author with either t_2 or t_3:

$$\texttt{SameAuthor}(t_1,\ t_2) + \texttt{SameAuthor}(t_1,\ t_3) = 1$$

Inter-task constraints: These are constraints that require decisions between multiple task to be consistent.
(3) Given two texts t_1, t_2, and user u: $\texttt{SameAuthor}(t_1, t_2) \wedge \texttt{AuthoredBy}(t_1, u) \Rightarrow \texttt{AuthoredBy}(t_2,\ u)$

(4) Given two users u, v, and text t: $\texttt{AgreeWith}(v, u) \wedge \texttt{AuthoredBy}(t,\ u) \Rightarrow \texttt{SupportedBy}(t,\ v)$

4. Empirical Evaluation

We evaluate the capability of our embedding model to reconstruct all relations between text and users in the social network. We have four prediction tasks, each of them focuses on one type of relation in the graph.

Debate.org: We run experiments on a dataset crawled from the debate.org website in June 2016. It contains user attributes (e.g., age, gender), stances on controversial issues, the debate text, and voting behavior by non-participating users. Our sample consists of 13,268 users with at least one attribute and 23,585 debates among them (with 60,132 debate rounds). There are 120264 examples of the AuthoredBy, 90720 SameAuthor, 17398 AgreeWith, and 45398 SupportedBy relations, respectively.

Experimental Settings: We used theano to implement the embedding neural network and Gurobi as our ILP solver. The embedding neural networks for both text and users contain two hidden layers and 300 hidden units in each layer. The vector from the last hidden layer is used as the

training	prediction	*prediction task*				
method	method	AuthoredBy	SameAuthor	VoteFor	SupportedBy	average
WE-baseline	local	50.99	61.62	57.88	52.27	55.69 (55.16)
LR-baseline	local	62.18	50.00	54.94	51.15	54.57 (55.85)
L_{TA}	local	65.53	64.74	58.08	53.51	60.47 (62.80)
L_{TT}		50.32	66.73	61.78	50.48	57.33 (56.51)
L_{AA}		49.90	61.21	65.50	49.00	56.40 (54.49)
$L(joint)$		64.08	66.68	63.64	55.28	62.42 (63.45)
L_{TA}	semi-joint	70.13	67.65	58.08	53.51	62.34 (65.79)
L_{TT}		50.68	70.48	61.78	50.48	58.36 (57.91)
L_{AA}		50.15	62.86	65.50	49.00	56.88 (55.15)
$L(joint)$		69.30	70.15	63.64	55.28	64.59 (66.90)
L_{TA}	joint	73.09	86.97	58.35	55.26	68.42 (73.80)
L_{TT}		50.77	88.81	60.99	50.86	62.86 (64.04)
L_{AA}		49.58	85.52	65.49	49.04	62.41 (62.41)
$L(joint)$		72.64	88.62	62.84	58.16	70.57 (74.91)

Table 1. Accuracy of Prediction Tasks Under Different Settings on Debate dataset.

embedding. For training, we used mini-batch gradient descent with early stopping based on the development set. All experiments use 5-fold cross validation, using one fold for development and three for training.

Input Representation: We used average word embedding as the text input. The concatenation of one-hot attribute vector, attribute words, and stance features are regarded as user input.

Tab 1 shows the accuracy on test set under different training and prediction methods. The reported results are averaged across 5 folds. We compare several training methods (first column). Our WE-baseline model is to simply use the average word (sentence) embedding to represent both text and attributes. LR-baseline model utilize logistic regression. It uses the concatenation of both input vectors as features, and considers the positive pairs as positive examples and vice versa.

All other training methods consider different losses or a combination of them. The prediction method (second column) includes several variations. *Local* refers to considering each prediction independently, while *semi-joint* (*joint*) enforces intra-task (and inter-task) constraints specified earlier. The final average column, reports the weighted average (by number of examples associated with each one of the prediction tasks) over the four prediction tasks.

The results reveal several interesting trends. First, we compare the impact of learning and using local vs. joint embedding. Local embedding, when used for predicting a task closely associated with the embedding objective, can help achieve a high accuracy. However, when we compared the averaged performance over *all* the prediction task, joint embedding always outperforms local embedding methods. These results are repeated when using either the local or joint prediction method. Second, joint prediction always outperforms local prediction. These results are repeated across all tasks. The best results are obtained using the joint prediction method over the joint embedding model.

5. Conclusions

We developed a model to embed users jointly in a latent representation, based on their social network information that includes attributes, interactions, and textual content. We considered both local and joint embeddings, as well as local and joint prediction methods that ensure predictions are consistent across the social network structure. In contrast to previous work, where there has been a clear divide between the social and language analysis, our results indicate that using a holistic approach produces consistent gains in predictive accuracy.

References

Amid, E. and Ukkonen, A. Multiview Triplet Embedding: Learning Attributes in Multiple Maps. In *ICML*, 2015.

Benton, A., Arora, R., and Dredze, M. Learning multiview embeddings of twitter users. In *ACL*, 2016.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *KDD*, 2016.

Huang, P., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2013.

Li, J., Ritter, A., and Jurafsky, D. Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks. *CoRR*, 2015.

Perozzi, B., R. Al-Rfou, Rami, and Skiena, S. Deepwalk: Online learning of social representations. In *KDD*, 2014.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. Line: Large-scale information network embedding. In *WWW*, 2015.

Wang, L., Li, Y., and Lazebnik, S. Learning deep structure-preserving image-text embeddings. *CoRR*, 2015.

Yang, Yi and Eisenstein, Jacob. Putting things in context: Community-specific embedding projections for sentiment analysis. *CoRR*, 2015.