A Generative/Discriminative Approach to De-construct Cascading Events

Sameera Horawalavithana
Department of Computer Science and Eng., University of South Florida
sameera1@mail.usf.edu

John Skvoretz
Department of Sociology, University of South Florida
jskvoretz@usf.edu

Adriana Iamnitchi
Department of Computer Science and Eng., University of South Florida
anda@cse.usf.edu

We introduce a generative/discriminative mechanism to predict the temporal dynamics of information cascade with the support of probabilistic models and Long-Short Term Memory (LSTM) neural networks. Our approach is to train a machine-learning algorithm to act as a filter for identifying realistic cascades for a particular social platform from a large pool of generated cascades. Our goal is to select the most realistic cascade with an accurate de-construction of user activity time-line. As an example in Twitter, we predict which user performs a retweet, and when she does such, in addition to the underlying cascade structure.

Probabilistic Cascade Generation

We employ a probability-based cascade generation approach to construct a pool of cascades given an initial seed (e.g., Reddit post, Tweet etc.). We used three conditional probability distributions drawn from the observed cascades in the training period. First distribution conditions the degree (i.e., number of adoptions), and the second distribution conditions the semantic values of the content (e.g., sentiment score of a Reddit comment) of an individual node by the level of the cascade tree. Third distribution conditions the sequence of adoption delays by the size of the cascade. We build the cascade trees recursively where the nodes are drawn at each level given the details from three conditional probability distributions.

Machine-Learning based Cascade Selection

First, we present the data model we used to represent cascades. A cascade consist of several nodes that branch together in a tree structure. Each node is described by its author (for example, the Reddit user who posts a message), post time and the content of the message. Nodes are ordered chronologically, by their post time.

LSTM-Model. For each individual to react in the cascade, the past reactions of predecessors matter. As an example, the last comment to a Reddit post could trigger an individual to make an immediate reaction. Meanwhile, such an individual needs to know the details of the overall conversation (i.e., How the conversation unfolded up until his reaction) We train a Long-Short Term Memory (LSTM) neural network by feeding cascades in a data model described earlier. We use the memory-cell design of a standard LSTM in our work. Cascades are different in shape, such that we feed cascades one by one to train in LSTM.

Prediction Tasks. In general, we predict the likelihood of observing a given sequence of adoptions in a cascade. We use two individual-level properties (e.g., branching factor and speed) of the cascade as the target units for the prediction tasks. In the first prediction task, we classify the messages as leaves (class 0) or branch (class 1) nodes in the tree. The second prediction task classifies messages by the delay with which they are posted in response to their parent. We refer to this delay as *propagation delay*. We consider the median propagation delay within a cascade as the borderline between the two classes: messages with a propagation delay larger than this median are called late adopters (class 1), while the others are early adopters (class 0).

Generative Test. We use the cascade generative approach described earlier to construct a thousand of realistic cascades given a particular initial seed. Specifically, the input to the generator is the original post or tweet by three sets of features:i) spatio-temporal properties, that capture the position of an individual message in a cascade; ii) user features; and iii) content features. Using the probabilistic generator, the model first outputs the structure of a cascade. We use an underlying diffusion network to associate node user information. As an example, we use the *shared-subreddit* network in Reddit and *follower* network in Twitter to overlay the generated cascade structure. Nodes' activation times are drawn from an empirically bench-marked propagation delay distribution.

Our object now becomes to select the best cascade according to the trained LSTM model. The trained LSTM models take as input the chronological sequence of the messages generated as part of the cascade but with all links in the cascade tree removed. The trained models will generate the labels that describe whether a message is a branch node or a leaf, and whether a message posted early or late compared with others. These labels are then compared with those generated probabilistically. We calculate accuracy as AUC and rank cascades by the mean accuracy. The best ranked cascade is the final solution for the given initial seed.

Evaluation

We empirically evaluate our solution on real data from three social media platforms: Reddit conversation, Twitter retweet chains and GitHub fork trees. We measure several key information spread measurements (e.g., structural virality, cascade lifetime), user-level measurements (e.g., user burstiness, new users over time), and content level measurements (e.g., content popularity). Our model outputs more accurate cascades with respect to multiple temporal dimensions (i.e., cascade structure and user involvement). Thus, it predicts cascading events at a finer granularity than what was previously attempted in aggregated measurements, such as the overall volume of user activities or the final size of a cascade.

MLNS'19, May 2019, Vermont USA
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn