The Missing Link: Link Prediction and Interest Adoption on Directed Social and Information Networks

Stephany Liu

December 12, 2016

1 Introduction

Tumblr is a popular social networking site that allows users to build a micro-blog and connect to other users by following each other, asking and answering questions, re-blogging other posts, and commenting on existing posts. Its unique format wherein each user is represented by a "mini-blog" website allows it to double as an information network, where each blog consists of a series of posts and "reblogs", or links to content by other users. In fact, such is the only way that links are publicly exposed on Tumblr: unlike peer sites such as Twitter, no public information is given about a user's followers and followees. Thus, Tumblr can be seen as a primarily content-based network.

However, behind the surface lies a rich social network that bonds users of all age groups from all over the world who do not necessarily have any relation in real-life. Over the years, Tumblr has become the medium of choice for many interest groups (colloquially known as "fandoms") to congregate and share their inspired creative works ("fanworks"). Other members of the platform focus on generating funny memes, editing impressive art collages, and discussing more serious issues, to the point where Tumblr activism is a colloquial term used to (derogatorily) refer to the social-justice movement that takes place on the site. These distinct communities, while informal, form interest groups over time that communicate primarily through the website's tagging feature.

In this project, I intend to explore two topics of interest: discovering missing links, which may represent potential friendships and missed connections rooted in mutual friends and shared interests, and predicting the adoption of interests by users over time. These questions have applications far beyond academic analysis: correctly predicting a missing link or interest, and communicating it to a user, can result in the budding of lifelong friendships and hobbies.

2 Related Work

The role of supervised learning in link detection has been investigated throughout the years, and different state of art techniques have been developed [5]. These fall roughly in three different categories: classification algorithms-based, probabilistic graph models, and matrix factorization. Among these options, the former allows for maximum experimenting with different features that may contribute to the success of a link prediction. One landmark study was performed by Hasan et al [2], comparing the performance of different machine learning classification algorithms on a link prediction problem. In an undirected graph based on a co-authorship network from the BIOBASE publication dataset, they measured three different types of predictors: proximity, node, and topological. They found that the most important features were keyword match count and sum of neighbors. The latter makes particular sense in the context of an undirected graph, in which directionality is not important and having higher degree increases the chances of an edge existing. The former is also strong for a formal context in which precise keyword tagging is expected.

For the link prediction problem, I continue their work in the context of the directed social and information network of Tumblr, taking inspiration from the classification algorithms that worked well for them, and adapting selected features that would make sense for a directed setting.

The second problem of interest adoption is similar in nature to another key issue in network research, community growth. Previously, Backstrom et al [3] studied factors that led to an individual joining communities on social networking site LiveJournal. Using the supervised learning technique of decision trees, they established the proportion of friends in a community who were also friends with each other as a critical factor to joining, suggesting that there is an advantage to having a stronger social support net when joining a community.

Adapting the findings of this study to the Tumblr community will be interesting, as the concept of "community" as represented by interest tags on Tumblr is a much more casual commitment than the process of joining a community on LiveJournal. Like Backstrom et al [3], I will apply supervised learning techniques to the prediction problem. By using more robust machine learning algorithms such as the random forest classifier and experimenting with network features in a directed graph setting where communities are only loosely defined, I hope to expand upon the quality of prediction and generalize the problem to more cases.

3 Methods

3.1 Dataset

To retrieve the dataset used for this project, I built multiple crawlers using Tumblrs developer API that extracts snapshots of each page visited, including a list of all blog posts, the users from whom these posts were reblogged, and the tag information associated with each post. Some of these crawlers followed a random walk model with teleports, where at each stage the crawler would either pick one of the links on the blog to follow or teleport (with probabiligy 0.15) to a random blog. Other crawlers built upon the existing dataset by exploring the neighbors of known nodes, thus expanding the network edgewise. In order to obtain as representative a subgraph of the entire active Tumblr community as possible, I executed these crawlers in alternating order over a period of many hours.

In setting the stage for the interest adoption problem, I repeated this process after a time interval of one month to obtain two distinct snapshots of the network one month apart. Henceforth, these two times will be referred to as t_1 and t_2.

3.2 Graph Model

Let $G = \langle V, E \rangle$ be a directed graph in which $v \in V$ represents a Tumblr blog and $e = (v_i, v_j) \in E$ represents a reblog by blog v_i from v_j visible within the first 20 posts of blog v_i at time t. This reblogged post establishes a visible link displayed on v_i's blog until it is replaced by more recent posts, allowing visitors to v_i to follow the link to v_j. True to Tumblr's nature as an information network, only the most recent posts on any Tumblr have link relevance, since a casual visit to a page would likely not expose content beyond the first 20 posts. The directed nature is significant, as it is not easy at all to discover rebloggers of a post from any particular blog, only from whom it has been reblogged.

3.3 Feature Selection

3.3.1 Link Prediction

For this problem, let v_a represent the source node and v_b the destination node of a prospective edge. The response is a binary variable indicating whether (v_a, v_b) exists in G.

As with any machine learning problem, selecting a representative feature set is one of the most critical steps of the process. This is especially true for the problem of link prediction [x], where classification algorithms are applied mostly out of the box, and interest adoption by extrapolation. In deciding which features to include, I consider both content-based features and network-based features.

Reciprocal Edge (`followed_by`) This is a binary value that is equal to 1 if v_b follows v_a and 0 otherwise. Reciprocity is often a key feature of social networks such as Tumblr, so if user A follows user B, user B is more likely to follow user A.

Node Degree (`in_deg_dst, out_deg_src`) I include the out-degree of node v_a and the in-degree of node v_b. Intuitively, the greater the number of edges linking out from i, the higher the chance of it forming an edge with a second node, and the greater the number of edges linking into j, the greater the chance that the second node will be j.

PageRank (`pr_src, pr_dst`) In a directed graph, the PageRank of a node i with out-degree d_i is given by

$$r_i = \sum_{i \to j} \beta \frac{r_i}{d_i} + \frac{1-\beta}{n}$$

For this dataset, I include both the PageRank score of the source and the score of the destination nodes. A node with higher PageRank is likely to receive more in-edges. On the other hand, a source with higher PageRank and low out-degree and a destination also with high PageRank and low-indegree might also indicate presence of an edge.

HITS (`hubs_src, auths_dst`) In a directed graph, the hub vector h and authority vector a are the eigenvectors of AA^T and A^TA, respectively. For node i, these values are h_i and a_i. Since hubs link to authorities, a source with high hub score and a destination with high authority score is a positive signal for an edge, while the inverse is a negative signal.

Shortest Path (`shortest_src_dst, shortest_dst_src`) The shortest path from v_a to v_b is given by the least number of hops necessary to reach v_b from v_a. Previous research [X, X] has demonstrated that most network nodes are connected in a very short distance, and existence of another short path from v_a to v_b is a positive signal for (v_a, v_b). I include both the shortest path from v_a to v_b and from v_b to v_a, since these values differ for a directed graph.

Triads (`triads_src, triads_dst`) For a node v_1, the number of triads is the number of distinct sets v_1, v_2, v_3 where v_2 and v_3 are other nodes in G and there exist pairwise links between all v_1 v_2, and v_3. For the purposes of this calculation, the graph is treated as undirected. I calculate this value for both v_a and v_b for inclusion, since research has shown that nodes in dense neighborhoods are likely to have more edges.

Clustering Coefficient (`cc_src, cc_dst`) For a node v, the clustering coefficient is given by

$$\frac{\#\text{ triads including } v}{\#\text{ triples centered around } v}$$

where v_2 and v_3 are other nodes in G and there exist pairwise links between all v_1 v_2, and v_3. Again treating the graph as undirected, I calculate this value for both v_a and v_b for inclusion for similar reasons to above.

Common Neighbors (`common_nbrs`) Measuring the number of shared out-links of v_a and v_b, this is given by the expression

$$|\Gamma(v_a) \cap \Gamma(v_b)|$$

Intuitively, if two users both reblog content from the same source, this indicates they likely have shared content that they are reblogging. Thus, it would be more likely for them to blog from each other as well.

Neighbor Similarity (`jac_nbrs, cos_nbrs`) Measuring the number of common neighbors is an absolute metric, but in a highly sparse dataset such as this one, it may be valuable to include more relative measures of similarity [X]. Here, I introduce two related but not identical measures. First, Jaccard similarity measures the ratio of overlap to union. For node neighbors, it is given by

$$\frac{|\Gamma(v_a) \cap \Gamma(v_b)|}{|\Gamma(v_a) \cup \Gamma(v_b)|}$$

Next, cosine similarity measures the angle between two vector representations in space. For vectors a and b, it is given by

$$\frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2 \sum_{i=1}^{n} b_i^2}}$$

In the context of this problem, let a and b be vectors where $a_i = 1$ if v_a has an edge to node i and 0 otherwise, and likewise for b_i. Since the numerator is nonzero only when $a_i = b_i = 1$, the above expression is simply equal to

$$\frac{|\Gamma(v_a) \cap \Gamma(v_b)|}{|\Gamma(v_a)||\Gamma(v_b)|}$$

I include both similarity scores as measures of relative similarity that can offer a more granular perspective than simply the magnitude of shared neighbors.

Tag Similarity (`jac_tags, cos_tags`) The measures here are similar to those introduced in the above paragraph, but operate on the domain of content, rather than network. Since many Tumblr blogs are rich multi-media content, the most easily accessible way to discern document content (treating each blog as a document) is to compile the tag information that each user adds to describe the posts on his/her blog. Let T_a and T_b denote the set of tags used by v_a and v_b. Then the Jaccard similarity score for the tags is given by

$$\frac{|T_a \cap T_b|}{|T_a \cup T_b|}$$

There are two main reasons why this measure alone is not very effective. First, each Tumblr user may tag an entity slightly differently ("legend of korra" and "the legend of korra" both refer to the popular animation series *The Legend of Korra*), and this measure will only match whole tag strings. Second, many users often embellish their tags with non-descriptive side commentary (e.g. "love this"), leading this value to skew low.

To address these and numerous other issues, I introduce the notion of TF-IDF (term frequency - inverse document frequency) from information retrieval. Term frequency for a blog is given by a vector of counts for every word, normalized. For a term t,

$$\text{TF}_t = \frac{\text{occurences of } t \text{ in tags}}{\text{num terms in doc}}$$

Inverse document frequency penalizes words that occur in more documents overall as being less significant, thus reducing the impact of auxiliary words or commentary (such as "love this"). For a term t,

$$\text{IDF}_t = \frac{\text{num docs}}{\text{num docs with } t}$$

Then the TF-IDF score of t is simply

$$\text{TF-IDF}_t = \text{TF}_t \cdot \text{IDF}_T$$

Using the above equation for cosine similarity, the similarity of two blog "documents" can be calculated. This gives a deeper measure of the similarity of content between two blogs.

3.3.2 Interest Adoption

Next, for the interest adoption problem, I predict based on the network at t_1 whether a node v that is not affiliated with interest i at t_1 would adopt i at t_2. I include only blogs that are not yet interested in i at t_1. The response variable is a binary indicator of whether v adopts i at t_2.

The interest tags chosen to study for this dataset are selected as a representative sample of the interests listed on Tumblr and recognized by site staff as popular [1].

Many of the predictors used in this problem are similar to above: PageRank, HITS scores, in-degree and out-degree, number of closed triads, clustering coefficient. I include a couple new measures: the number of neighbors who already have the interest at t_1, the number of neighbors in the interest at t_1 who are also linked to each other, and the proportion of the former who have the latter connection.

3.4 Classification Algorithms

I consider a range of classification algorithms, each with individual strengths over different datasets. These will be mainly used out of the box, so implementation details are not especially noteworthy. A summary of methods used are is given blelow.

Tree Ensembles Considered to be state of the art in performance, tree-based methods excel at detecting complex, non-linear decision boundaries. While single decision trees, as used by Backstrom et al [3], can serve as powerful tools for visualizing the relationships between predictors, they are often prone to overfitting. Tree ensembles address this issue by aggregating the fit of many different trees to reduce variance. I consider three different variants.

Bagging Short for bootstrap aggregating, this approach treats the dataset as a population from which to resample (with replacement) repeatedly. A separate decision tree is fitted for each sample, and the results are aggregated at the end, thus reducing the risk of any single particular tree overfitting.

Random Forest A further improvement upon bagging trees, random forests further reduce variance by randomly limiting the number of predictors considered at each split of a decision tree. This works because decision trees employ a greedy algorithm to reduce error at each split, which leads to correlated trees (inflating variance).

Gradient Boosting Trees are grown sequentially, using information from the previous fitted tree. Each tree is fitted to the current residuals, then shrunken by a parameter, allowing the model to learn the data slowly and improve in areas in which preceding trees were less strong.

Support Vector Machines Another popular classic approach to classification, SVMs apply a maximal margin classifier to high-dimensional space. They are generally quite good out of the box. I consider both linear kernels and radial kernels to account for different decision boundaries.

Nearest Neighbors Nearest neighbors classifies based on distance to existing points in the dataset without any assumptions about true distribution. It serves as a solid baseline for comparison against the more advanced algorithms presented above.

Naive Bayes A quick and simple classifier, Naive Bayes makes the assumption of independent predictors, which does not hold for this dataset. However, as a classic classification algorithm, I include it as another baseline for comparison.

3.5 Evaluation

Cross-Validation 10-fold cross-validation is a reliable technique to evaluate accuracy against a held-out set and estimate the test error. For link prediction and interest adoption, this will be calculated as the accuracy at a threshold of 0.5.

Area Under ROC-Curve To test for model validity across different thresholds of accuracy and visualize the tradeoff between precision and recall, ROC curves will be plotted, and the area under will be calculated as a measure of the global soundness of a model.

4 Results & Discussion

4.1 Link Prediction

The tree ensemble classification techniques, popular for their ability to reduce variance through use of multiple classifiers, demonstrate superior accuracy. The highest scoring model was a random forest classifier considering 8 predictors at each split, with a score of 92.39%. This was followed extremely closely by bagging ensembles at 92.29%.

The remaining methods drop off in accuracy: boosting (5000 trees) performs comparably with

Classifier	Accuracy	AUC
Random Forest	0.9238523	0.9796161
Bagging	0.9228807	0.9786625
Boosting	0.8906971	0.9626868
SVM (Linear kernel)	0.8690794	0.8690794
SVM (radial kernel)	0.8814671	0.8814671
5-NN	0.8299733	0.9004024
Naive Bayes	0.7787224	0.9763964

Figure 1: Summary of scores by classification algorithms

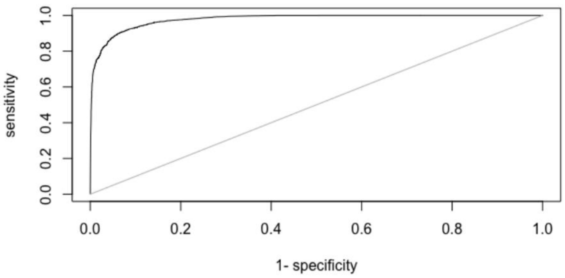

Figure 2: ROC curve of random forest classifier

an accuracy of 89.07%. SVM with a linear kernel, kNN (k = 5), and Naive Bayes are significantly less accurate. The latter, with just 77.87%, exhibits particularly poor performance. Given that one of the base assumptions of Naive Bayes is the independence of features, and topological network features are highly correlated, this is not at all surprising. Naive Bayes is also likely too simple of a model to capture the complexity of the data.

The dataset used for this problem was approximately equally split between the two classes, which allows accuracy to be a more reliable measure of performance. However, the AUC scores can provide a more complete picture of the tradeoff in performance. Since all AUC scores are relatively high and within a narrow range, globally it seems the models are reliable.

To visualize the tradeoff, Figure 2 include an image of the ROC curve for the random forest classifier. The tradeoff in precision and recall depends on the nature of the problem. For a link prediction problem such as this which has the end goal of recommending blogs to follow, one might consider the cost of a false negative (a missed connection with a Tumblr blog that contains enriching fandom content, and potential friendship with the user behind the blog) to far outweigh the cost of a false positive (slight amount of wasted time, often less than a minute, browsing a Tumblr blog that isn't relevant to a user's interest). In addition, in a sparse social network setting such as this, the existing blog connections often only represent a small percentage of the total possible connections that may potentially exist, due to factors such as the extremely high volume of blogs within each interest, limited time spent by users browsing the social network to create connections, and the limits of crawling using the Tumblr API, which retrieves only 20 posts (edges) at a time. In light of this, high recall should be be favored over high precision.

According to the random forest classifier, the most important predictors include `hub_score`, `followed_by`, `in_degree_dst`, `pr_dst`, and `shortest_src_dst`. They are scored across two measures: the mean decrease in accuracy is the decrease in model accuracy without the predictor, and the mean decrease in GINI measures the average decrease in node purity with exclusion of the predictor. The strength of the directed graph

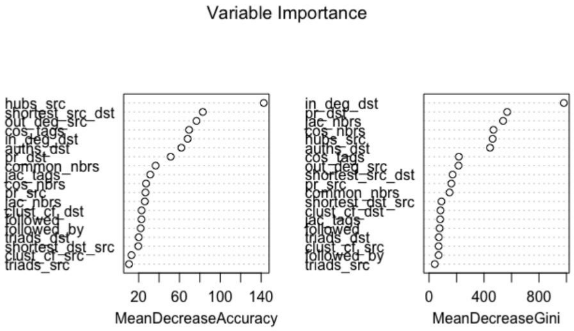

Figure 3: Importance of predictors in link prediction

score variables confirm our hypothesis that level of influentiality in the network as a whole are key to link prediction. The importance of the shortest paths measures also validate conclusions made in earlier research [2].

4.2 Interest Adoption

The SVM with radial kernel achieves the highest accuracy for the interest adoption problem, with a score of 77.87%. Following very closely behind are boosting (5000 trees) and SVM with linear kernel, with accuracy of 77.54% and 77.37%, respectively. The random forest classifier also performs very well at 77.04%. As in link prediction, the worst performing model is Naive Bayes, with a dismal accuracy of 59.90%. The contrast in accuracy is even more stark this time due to the highly correlated nature of the variables: for example, between `nbrs_interest` (number of neighbors of a node with the interest in question at t_1), `num_nbrs_interest_linked` (number of neighbors of a node with the interest who are also linked directly to each other), and `prop_nbrs_interest_linked` (the proportion of a node's neighbors who are also linked to each other). Similar to link prediction, the dataset for this classification was also roughly evenly split between the two classes.

Classifier	Accuracy	AUC
Random Forest	0.7703827	0.8404727
Bagging	0.7520799	0.8213748
Boosting	0.7753744	0.8411371
SVM (Linear kernel)	0.7737105	0.8187889
SVM (radial kernel)	0.7787022	0.8346254
5-NN	0.6688852	0.7236484
Naive Bayes	0.5990017	0.8213748

Figure 4: Summary of scores by classification algorithms for interest adoption

The most important features, as shown in Figure 5, are `nbrs_interest`, `category`, and `hub`, with the former carrying the most impact by far on the model accuracy. It is clear that the most influential factor on a user adopting an interest is the number of blogs he/she follows that contain content related to this interest. However, one must be careful to not assume causation. Perhaps this user is drawn in by the volume of content related to this interest on his dashboard (produced by the blogs he/she follows). Or perhaps this user originally decided to follow these blogs because they blog content in related domains, and it is only logical to make the jump between interests. For a concrete example, take a fan of the *Harry Potter* series who follows book-themed blogs. These book-themed blogs might post content both about *Harry Potter* and *The Mortal Instruments*, another popular young-adult fantasy novel series. Just as the owners of these blogs came to enjoy both series, this user may also find *The Mortal Instruments* interesting as a *Harry Potter* fan. Thus, the influence of the number of neighbors with the interest at t_1 suggests a strong correlation with the user then adopting the interest at t_2, but causation cannot be established.

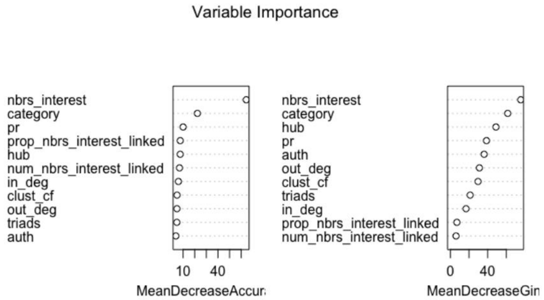

Figure 5: Importance of predictors in interest adoption

The influence of the predictor `category` confirms the hypothesis that interest adoption varies across different mediums. This makes sense intuitively, as the rate of adoption and social factors for following an animated television series are not the same as those for a celebrity or a social justice movement.

That the variable `hubs` is also a strong predictor for interest adoption demonstrates the tendency of most Tumblr blogs to take on either a hub role or an authority role; higher hub scores are seen among blogs that tend not to adopt interests. An "authority" blog, with large followings, may create a lot of original art and content related to its interests to distribute to its followers. This curation process involves careful tagging of posts, so the content can be more easily found using website search functions. Additionally, to maintain quality for the large following, an "authority" blog generally needs to particular about the content that goes onto the blog, ensuring that it is relatively thematic and thus appealing to digest. A "hub" blog, on the other hand, would tend to reblog posts made by the "authority" blogs, producing a hodgepodge of content for the user's own enjoyment but not curated for followers. As a result, the user may not bother to tag very thoroughly, if at all, and can be much more liberal about posting content from different interest domains on his blog. This greatly expands the potential interests in a snapshot of the blog (20 posts), and thus it is much less likely that such a blog would contain content about any particular interest at a given point in time.

Unlike Backstrom et al [3], I found neither the `num_nbrs_interest_linked` (number of neighbors of a node with the interest who are also linked directly to each other), and `prop_nbrs_interest_linked` (the proportion of a node's neighbors who are also linked to each other). The vast majority of these values were 0 in for this network for both prediction classes, rendering these features useless in differentiation. The difference may be attributed to the sparse nature of the Tumblr network. Each Tumblr blog can contain only a limited number of posts (and thus out-edges) at any given point in time viewed, and as a hybrid social-information network of miniature sites, this is the way in which Tumblr was designed to be browsed.

This drastically decreases the density of edges and content in the Tumblr network, compared to even the sparse LiveJournal social network analyzed by Backstrom et al. Furthermore, while communities are defined entities on LiveJournal and collectively play an integral part in the site experience, they are much less defined on Tumblr, not being formally integrated into the site architecture and merely being formed by users around tags (later recognized by site staff as being "communities" or "fandoms") [1]. This informal attitude towards communities leads to more transient and holistically lower participation compared to LiveJournal, where community membership is taken more seriously. Thus, it is less likely for users to be linked simply because they are members of the same ”community” on Tumblr.

5 Conclusion & Future Work

In conclusion, through the use of supervised learning classification algorithms over the Tumblr network dataset, one can see the importance of influence in predicting the existence of a link, particularly in a social network such as Tumblr. The random forest classifier continues to show dominance. With the addition of thorough content-based similarity measures, I have improved upon the prediction strength shown in previous landmark studies [2, 4].

In comparison to link prediction, the issue of interest adoption was a relatively original problem to explore, particularly in the context of Tumblr, where links are only loosely defined. Although the overall accuracy for all methods is lower than for link prediction, this should not be a surprise: after all, the basis of interest adoption is based solely on the presence of keywords in the tags of blog posts. This can often become very messy, since many users adopt various abbreviations for the same interest, and each blog can only be trusted to be consistent with itself. To further explore this topic, it would be interesting to delve into natural language processing techniques such as entity linking and integrate them into the context of the network.

These techniques and others are outside the scope of this project, but may prove interesting to explore in the future. Here, I considered the existence of a link from a blog to another blog as a single link, but this can easily be converted into edge weights. For example, a blog containing multiple links to another blog demonstrates greater favor towards that blog. Such networks with weighted edges would be interesting to consider in future research.

References

[1]

[2] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

[3] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’06, pages 44–54, New York, NY, USA, 2006. ACM.

[4] William Cukierski, Benjamin Hamner, and Bo Yang. Graph-based features for supervised link prediction. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1237–1244. IEEE, 2011.

[5] Peng Wang, BaoWen Xu, YuRong Wu, and Xiaoyu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.