Predicting Component Utilities for Linear-Weighted Hybrid Recommendation

Fatemeh Vahedian & Robin Burke
Center for Web Intelligence, Depaul University, Chicago, IL 60604
{fvahedia, rburke}@cs.depaul.edu

ABSTRACT

An effective technique for recommendation in social media and other heterogeneous networks is the weighted hybrid of low-dimensional components (WHyLDR). Recent studies have shown this technique is comparable to other integrative approaches while being considerably more flexible. One key issue for the implementation of a WHyLDR system is the choice of components to generate. Research has shown that the contribution of components based on different network paths varies in unexpected and domain-dependent ways. This work examines an information theoretic technique for estimating component performance. Using a real-world social media dataset, we show that this technique is useful both for optimization (estimating component weights) and for determining which components to include in a hybrid.

1. INTRODUCTION

Social media sites are an important element of today's Internet, drawing millions of users each day. The wealth of information found in these sites makes recommender systems essential. Such sites often must integrate recommendations of many types: recommending content, like-minded users or appropriate tags to name just a few possibilities. Our approach, called the Weighted Hybrid of Low-Dimensional Recommenders (WHyLDR), is designed to support the flexible creation and rapid deployment of a wide variety of recommenders in a heterogeneous environment. We have demonstrated its effectiveness in prior work focusing on social tagging systems [8, 7, 6, 3, 4].

A social media site can be viewed as a heterogeneous network defined by a diversity of objects and relations. This diversity gives rise to a wide variety of recommendation tasks. For example, consider the popular social media site Yelp, which allows users to review and rate restaurants and other types of businesses.[1] In Yelp, there are users, reviews, businesses, and other related elements. One obvious recommendation task within Yelp is to recommend new businesses to users, but there are a variety of others. Recommending other users to befriend, recommending locations, and recommending categories of businesses are all user-focused recommendation tasks. A site like Yelp may also be interested in recommending users to businesses for marketing purposes. In addition, a user may wish to constrain the recommendations in various ways: looking for a recommended business in a particular category or in a particular location, for example.

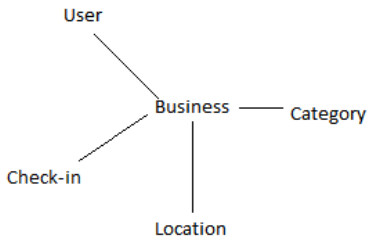

Figure 1: Yelp network schema

As in other work with heterogeneous networks [10], the WHyLDR model views the network structure as a set of mappings, or projections from nodes to nodes: $proj_A(n) \rightarrow \{m_0, m_1, ..., m_i\}$ where A is a set of paths, n is a starting node and the ms are nodes reachable from n via some path in A. Sets of paths that pass from one type of node to another over specific categories of edges are known as meta-paths.

Meta-paths can also be envisioned as paths through a network *schema*, an abstract view of a heterogeneous network that shows the node types and the possible connections between them. Figure 1 shows the schema for part of the Yelp social network. In this network, a path from any user node to the businesses that individual has reviewed to the categories associated with that business would be a UBC metapath. Following such a path for any given user yields a type of user profile – in this case, a user represented as the categories of rated businesses. Using such profiles, we can build standard collaborative filtering components to select neighbors of users and generate recommendations. Out of a collection of such meta-paths, we can build an ensemble of components, each capturing a different aspect of the network.

[1] www.yelp.com

Proceedings of the 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014), collocated with ACM RecSys 2014, 10/06/2014, Foster City, CA, USA.

While this model has proved successful in multiple social media settings, a key problem remains of how to control hybrid size. The set of components, like the set of possible meta-paths, is unbounded. We have found that in some settings components built using longer paths can outperform those using shorter paths. There is therefore no simple way to choose a limited number of components and be sure of optimal performance. In this paper, we examine an information gain metric that can be used to estimate the potential contribution of components. We show that this metric can be used in two ways:

- to estimate the hybrid weights directly, and
- to filter components based on their expected contribution to the hybrid.

2. WEIGHTED HYBRID

A weighted hybrid recommender is a system comprised of multiple recommendation components, each of which returns a real-valued score for a combination of user and item. The scores from all the components are combined in a weighted sum [2]. More formally,

$$s(u,i) = \sum_j \alpha_j s_j(u,i)$$

where $s(u,i)$ is the overall score computed for a user-item combination, $s_j(u,i)$ is the score computed by the jth component, and α_j is the weight associated with the jth component.

The weights are learned through an optimization procedure as discussed below. The components are a function of the recommendation task and the structure of the network.

WHyLDR components are built from two-dimensional matrices familiar to researchers in collaborative recommendation [5]. A user-based matrix is one in which the rows are users and the columns are the destination nodes for a given meta-path. Users are compared on the basis of their profiles and peer users form a neighborhood from which a target user's preferences for unknown items can be extrapolated.[2]

The experiments reported here are for the task of recommending businesses to users. Four types of recommendation components are included in our Yelp model:

- user-based KNN components constructed based on meta-paths starting from a user node,
- item-based KNN components from meta-paths starting from a business node,
- cosine components in which two separate matrices are constructed (one for users and one for items) and then items and users are compared directly based on their profiles (the rows in their respective matrices), and
- a non-personalized recommendation component based solely on item popularity.

[2]All of the optimizations that have been applied to collaborative recommenders can therefore be applied to the individual WHyLDR components, for example, matrix factorization. We plan to study the properties of such optimizations on our hybrids in future work.

Table 1: Meta-paths for recommendation components

Type	Meta-paths
User-based	UB, UBC, UBL, UBH, UBCB, UBLB, UBHB
Item-based	BU, BL, BC, BH, BLBC, BLBU, BUBU, BUBL
Cosine	UBC, UBH

3. CONTROLLING META-PATH GENERATION

There is no requirement that meta-paths be simple: nodes and edges can be revisited, as seen in components like kNN_{UBLB}, where the meta-path loops from businesses to locations and back to businesses again. Component generation could in theory continue indefinitely. However, there are significant computational costs in generating components and in optimizing a hybrid with a large number of components. Moreover, some components will make only a minor contribution to recommendation performance. It is therefore important to control this process. Ideally, we would like to be able to estimate in advance what components are likely to make a substantial contribution to the hybrid and make an informed decision to trade off expected accuracy against the computational costs of additional components.

We have developed a measure based on mutual information for each meta-path to estimate the utility of recommendation components. For a given two-dimensional projection AB, the mutual information can be calculated as

$$I(A,B) = H(A) - H(A|B)$$

where $H(A)$ is entropy of dimension A and $H(A|B)$ is the conditional entropy. Entropy is defined as

$$H(A) = -\sum_i p(a_i) log(p(a_i))$$

The entropy is therefore a function of the probability of occurrence of nodes in each dimension. In our networks, we define probability of node a_i from dimension A based on the node degree:

$$p(a_i) = \frac{Degree(a_i)}{\sum_i Degree(a_i)}$$

The intuition behind this measure comes from results related to random walks. As the length of a random walk approaches infinity, the probability of hitting a node converges to its normalized degree [**?**].

Conditional entropy measures the uncertainty of one dimension given another dimension, computed by summing up individual conditional probabilities. In our networks, we define the conditional probability $P_M(b_i|a_j)$ as the fraction of meta-paths of type M leaving node a_i and arriving at b_j out of all meta-paths of type M.

$$P_M(b_i|a_j) = \frac{|a_j \underset{M}{\longrightarrow} b_i|}{\sum_k |a_j \underset{M}{\longrightarrow} b_k|}$$

For example, consider the user-business-category meta-path and associated recommendation component. The values for $H(U)$ and $H_{UBC}(U|C)$ can be calculated using the formulas above. If these values were roughly the same then the $I_{UBC}(U, C)$ will be around zero. This suggests that the meta-path does not add much information beyond what is already contained in the U dimension and that the UBC meta-path is unlikely to give rise to a useful recommendation component. The same principle can be applied to any user-based or item-based component. For the cosine components, we used the minimum information gain from either constituent meta-path. For the popularity component, we used minimum information gain across all components.

In our previous experiments, we found that there was a significant correlation between the information gain associated with a meta-path and the learned α weight for a component built using that meta-path [4]. In this work, we sought to build on that result by substituting our measure of meta-path information for weights learned through optimization. We use a simple normalization of the information gain measure, so the contribution of ith component can be calculated as:

$$\alpha_i = \frac{InformationGain(i)}{\sum_{j=1}^{k} InfomationGain(j)} \quad (1)$$

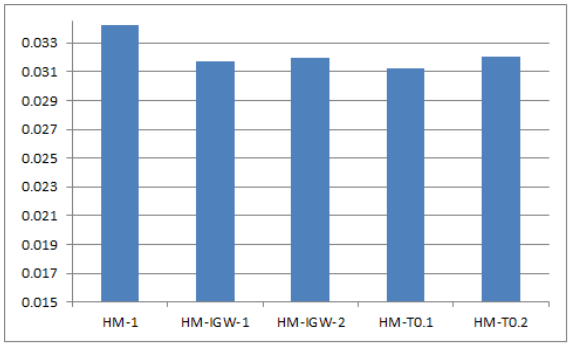

Figure 2: F1 values for each hybrid model

Another way to use the information gain is as a filter controlling which components are incorporated into the hybrid. In these experiments, we used two different thresholds (0.1 and 0.2) for rejecting components with low information gain. Table 2 shows which components were dropped in each case: the only difference between the lower threshold and the higher one was the exclusion of the BUBU component at the higher threshold.

The results below report on four different configurations of the system: HM-1, a hybrid model including all the components discussed in Section 2 with the weights learned through optimization; HM-IGW, a hybrid including the same components as HM-1 but with weights calculated as normalized information gain; HM-T0.1 and HM-T0.2, hybrids with learned weights but components with low information gain filtered out using thresholds of 0.1 and 0.2, respectively. Table 2 shows which components were removed in each case.

3.1 Methodology

For the experiments reported here, we used the **Yelp** Academic Dataset, and followed the four fold cross validation methodology described in [4]. The α weights were learned using Particle Swarm Optimization (PSO) [9] with recall as the optimization objective. In order to compare hybrid model performance, we calculate recall and precision at list sizes from 1 to 10. These are averaged for each partition and then averaged over all partitions. We also calculate F_1 at list size 10, the harmonic mean of precision and recall. We also measured the diversity of the results returned by these recommendations, but omit these results for reasons of space.

4. RESULTS

Figure 3 shows the recall-precision curve for the top individual recommendation component (kNN_{UB}) and the four hybrids. There are several points to note here. One is there is a relatively small difference between the performance of hybrid with learned weights HM-1 (dashed line) and the hybrid with weights estimated from our information gain measure HM-IGW (solid line with diamond marks) and indeed, some data points of the HM-IGW curve are above those of the learned hybrid. The results for the thresholded hybrids (HM-T0.1 and HM-T0.2) are generally lower except at list size of 1 and list size of 10.

Figure 2 shows a different perspective on these experiments with the F_1 measure computed at a recommendation list size of 10. Again, the hybrid with the learned weights shows the top performance with the other hybrids about 6-7% below.

What we find therefore is that there is a modest trade-off between recommendation performance with a fully-optimized hybrid and with weights computed using information gain. As the number of components increases the dimensionality of the optimization space increases, making the optimization step more time-consuming and more prone to over-fitting. Information gain can be computed directly from the meta-path expansions used to create the recommendation components and so is essentially free.

We also see that information gain may be useful in controlling the size of the hybrid. HM-T0.2 has one third fewer components than the full hybrid and approximately 6% lower F_1 performance. This suggests a process whereby meta-paths are generated and their information gain assessed before components are constructed, thus saving both off-line optimization time and run-time computation.

5. CONCLUSIONS

A key challenge in social media recommendation is to integrate the different types of information available in such systems to enhance recommendations and to offer recommendations of multiple types. The WHyLDR approach has been demonstrated to be successful in both of these respects. As there are an unbounded number of possible components in a WHyLDR hybrid, the questions arise of how to choose components for a hybrid and when to stop adding to it. In this paper, we show that our information gain measure shows promise for controlling hybrid creation and possibly for estimating component weights. While the full hybrid with learned weights showed the strongest performance, the

Table 2: Removed components at each threshold

Threshold	Components removed
0.1	UBH, UBHB, BLBC, BUBC, BUBL
0.2	Above, plus BUBU

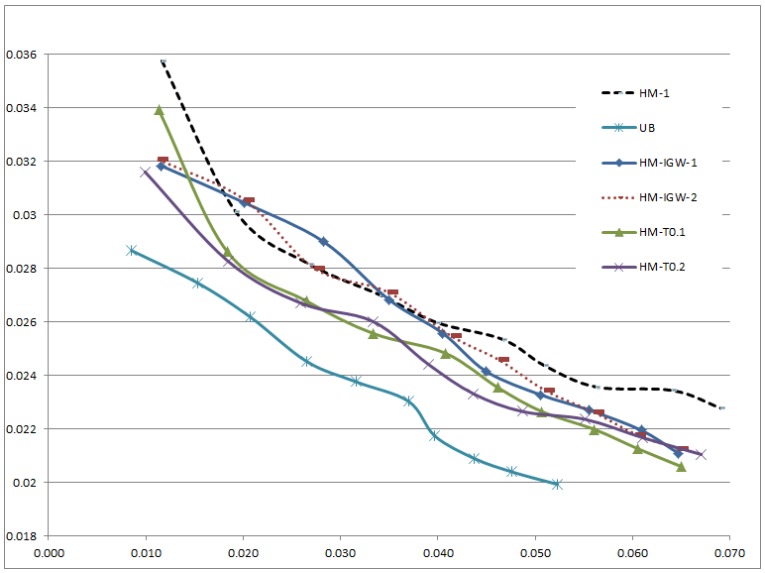

Figure 3: Recall vs. Precision

versions where our information gain heuristic was applied offered comparable results at lower computational cost.

One of the key problems unaddressed by this work is the influence of recommendation task. We know from prior work that, for most recommendation problems, the most strongly-contributing component is the one that directly maps to the recommendation task. For example, in this paper, kNN_{UB} is the most important single component because we are recommending businesses to users. If on the other hand, we were recommending locations, we would expect the kNN_{UBL} component to be a stronger contributor. This effect is not captured by our information gain measure, which is a function of the network structure alone. In future work, we will examine additional recommendation tasks and try to determine how to modify our information gain measure to take the task into account. We will also be experimenting with other heterogeneous network data sets to understand the generalizability of these results.

6. REFERENCES

[1] https://www.yelp.com/academic _dataset.

[2] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[3] R. Burke and F. Vahedian. Social web recommendation using metapaths. In *RSWeb@RecSys*, 2013.

[4] R. Burke, F. Vahedian, and B. Mobasher. Hybrid recommendation in heterogeneous networks. In *Proceedings of the 22nd Conference on User Modeling, Adaptation and Personalization*, 2014, to appear.

[5] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer, 2011.

[6] J. Gemmell, T. Schimoler, B. Mobasher, and R. Burke. Recommendation by example in social annotation systems. *E-Commerce and Web Technologies*, pages 209–220, 2011.

[7] J. Gemmell, T. Schimoler, B. Mobasher, and R. Burke. Tag-based resource recommendation in social annotation applications. *User Modeling, Adaption and Personalization*, pages 111–122, 2011.

[8] J. Gemmell, T. Schimoler, B. Mobasher, and R. Burke. Resource recommendation in social annotation systems: A linear-weighted hybrid approach. *Journal of Computer and System Sciences*, 78(4):1160–1174, 2012.

[9] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[10] Y. Sun and J. Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2012.