Similarity-Weighted Association Rules for a Name Recommender System

Benjamin Letham

Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA, USA
bletham@mit.edu

Abstract. Association rules are a simple yet powerful tool for making item-based recommendations. As part of the ECML PKDD 2013 Discovery Challenge, we use association rules to form a name recommender system. We introduce a new measure of association rule confidence that incorporates user similarities, and show that this increases prediction performance. With no special feature engineering and no separate treatment of special cases, we produce one of the top-performing recommender systems in the discovery challenge.

Keywords: association rule, collaborative filtering, recommender system, ranking

1 Introduction

Association rules are a classic tool for making item-based recommendations. An association rule "$a \rightarrow b$" is a rule that item(set) a in the observation implies that item b is also in the observation. Association rules were originally developed for retail transaction databases, although the same idea can be applied to any setting where the observations are sets of items. As part of the ECML PKDD 2013 Discovery Challenge, in this paper we consider a setting where each observation is a set of names in which the user has expressed interest. We then form association rules "$a \rightarrow b$," meaning that interest in name a (or, in general, set of names a) implies interest in name b. The strength with which a implies b is called the *confidence* of the rule, and in Section 2.2 we explore different measures of confidence.

Association rules provide an excellent basis for a recommender system because they are scalable and interpretable. The scalability of association rule algorithms has been well studied, and is often linear in the number of items [1]. Using rules to make recommendations gives a natural interpretability: We recommend name b *because* the user has expressed interest in name a. Interpretability is an important quality of predictive models in many contexts, and is especially important in recommender systems, where it has been shown that providing the user an explanation for the recommendation increases acceptance and performance [2, 3].

One of the most successful tools for recommender systems, particularly at a large scale, is collaborative filtering [4, 5]. Collaborative filtering refers to a large class of methods, of which here we focus on user-based collaborative filtering and item-based collaborative filtering [6]. In user-based collaborative filtering, recommendations are made by finding the most similar users in the database and recommending their preferred items. In item-based collaborative filtering, similarity is measured between items and the items most similar to those already selected by the user are recommended. Like association rules, collaborative filtering algorithms generally have excellent scalability.

Our main contribution is to use ideas from collaborative filtering to create a new measure of association rule confidence, which we call *similarity-weighted adjusted confidence*. We maintain the excellent scalability and interpretability of collaborative filtering and association rules, yet see a significant increase in performance compared to either approach. Our method was developed in the context of creating a name recommender system for the ECML PKDD 2013 Discovery Challenge, and so we compare the similarity-weighted adjusted confidence to other collaborative filtering and association rule-based approaches on the Nameling dataset released for the challenge.

2 Similarity-Weighted Association Rule Confidence

We begin by introducing the notation that will be used throughout the rest of the paper. Then we discuss measures of confidence, introduce our similarity-weighted adjusted confidence, and discuss strategies for combining association rules into a recommender system.

2.1 Notation

We consider a database with m observations $x_1, \ldots, x_m$, and a collection of n items $Z = \{z_1, \ldots, z_n\}$. For instance, it may be m visitors to a name recommendation site, with Z the set of valid names. Each observation is a set of items: $x_i \subseteq Z, \forall i$. We denote the number of items in x_i as $|x_i|$.

We will consider rules "$a \rightarrow b$" where the left-hand side of the rule a is an itemset ($a \subseteq Z$) and the right-hand side is a single item ($b \in Z$). Notice that a might only contain a single item. We denote as $\mathcal{A}$ the collection of itemsets that we are willing to consider: $a \in \mathcal{A}$. One option for $\mathcal{A}$ is the collection of all itemsets, $\mathcal{A} = 2^Z$. If Z is very large this can be computationally prohibitively expensive and some restriction may be necessary. In our experiments in Section 3 we took $\mathcal{A} = Z$, that is, all itemsets of size 1.

2.2 Confidence and Similarity-Weighted Confidence

The standard definition of the confidence of the rule "$a \rightarrow b$" is exactly the empirical conditional probability of b given a:

$$\text{Conf}(a \rightarrow b) = \frac{\sum_{i=1}^{m} \mathbb{1}_{[a \subseteq x_i \textbf{ and } b \in x_i]}}{\sum_{i=1}^{m} \mathbb{1}_{[a \subseteq x_i]}}, \tag{1}$$

where we use $\mathbb{1}_{[\text{condition}]}$ to indicate 1 if the condition holds, and 0 otherwise.

This measure of confidence corresponds to the maximum likelihood estimate of a specific probability model, in which the observations are i.i.d. draws from a Bernoulli distribution which determines whether or not b is present. Because of the i.i.d. assumption, all observations in the database are considered equally when determining the likelihood that a implies b. In reality, preferences are often quite heterogeneous. If we are trying to determine whether or not a new user x_ℓ will select item b given that he or she has previously selected itemset a, then the users more similar to user x_ℓ are likely more informative. This leads to the *similarity-weighted confidence* for user x_ℓ:

$$\text{SimConf}(a \to b | x_\ell) = \frac{\sum_{i=1}^{m} \mathbb{1}_{[a \subseteq x_i \textbf{ and } b \in x_i]} \text{sim}(x_\ell, x_i)}{\sum_{i=1}^{m} \mathbb{1}_{[a \subseteq x_i]} \text{sim}(x_\ell, x_i)}, \tag{2}$$

where $\text{sim}(x_\ell, x_i)$ is a measure of the similarity between users x_ℓ and x_i. The similarity-weighted confidence reduces to the standard definition of confidence under the similarity measure $\text{sim}(x_\ell, x_i) = 1$, as well as

$$\text{sim}(x_\ell, x_i) = \begin{cases} 1, & \text{if } x_\ell \cap x_i \neq \varnothing. \\ 0, & \text{otherwise.} \end{cases}$$

Giving more weight to more similar users is precisely the idea behind user-based collaborative filtering. A variety of similarity measures have been developed for use in collaborative filtering, one of the more popular of which is the cosine similarity, which we use here:

$$\text{sim}(x_\ell, x_i) = \frac{|x_\ell \cap x_i|}{\sqrt{|x_\ell|}\sqrt{|x_i|}}. \tag{3}$$

2.3 Bayesian Shrinkage and the Adjusted Confidence

In [7], we show how the usual definition of confidence can be improved by adding in a beta prior distribution and using the maximum *a posteriori* estimate. The resulting measure is called the *adjusted confidence*:

$$\text{Conf}_K(a \to b) = \frac{\sum_{i=1}^{m} \mathbb{1}_{[a \subseteq x_i \textbf{ and } b \in x_i]}}{\sum_{i=1}^{m} \mathbb{1}_{[a \subseteq x_i]} + K}, \tag{4}$$

where K is a user-specified amount of adjustment, corresponding to a particular pseudocount in the usual Bayesian interpretation. In particular, the adjusted confidence is equivalent to there being an additional K observations containing a, none of which contain b. This reduces the confidence of "$a \to b$" by an amount inversely proportional to the support of a, allowing low-support-high-confidence rules to be used in the computation, but giving more weight to those with higher support. In terms of the bias-variance tradeoff, adjusted confidence leads to an increase in performance by reducing the variance of the estimate for itemsets

with low support. The Nameling dataset used here is quite sparse, so we add the same adjustment to our similarity-weighted confidence, producing the *similarity-weighted adjusted confidence*:

$$\text{SimConf}_K(a \to b|x_\ell) = \frac{\sum_{i=1}^m \mathbb{1}_{[a \subseteq x_i \textbf{ and } b \in x_i]} \text{sim}(x_\ell, x_i)}{\sum_{i=1}^m \mathbb{1}_{[a \subseteq x_i]} \text{sim}(x_\ell, x_i) + K}. \tag{5}$$

When $K = 0$, this reduces to the similarity-weighted confidence in (2).

2.4 Combining Association Rules to Form a Recommender System

The similarity-weighted adjusted confidence provides a powerful tool for determining the likelihood that $b \in x_\ell$ given that $a \subseteq x_\ell$. In general there will be many itemsets a satisfying $a \subseteq x_\ell$, so to use the association rules as the basis for a recommender system we must also have a strategy for combining confidence measures across multiple left-hand sides. For each left-hand side $a \in \mathcal{A}$ satisfying $a \subseteq x_\ell$, we can consider $\text{SimConf}_K(a \to b|x_\ell)$ to be an estimate of the probability of item b given itemset x_ℓ. There is a large corpus of literature on combining probability estimates [8, 9], from which one of the most common approaches is simply to compute their sum. Thus we score each item b as

$$\text{Score}(b|x_\ell) = \sum_{\substack{a \subseteq x_\ell \\ a \in \mathcal{A}}} \text{SimConf}_K(a \to b|x_\ell). \tag{6}$$

A ranked list of recommendations is then obtained by ranking items by score.

A natural extension to this combination strategy is to consider a weighted sum of confidence estimates. We consider this strategy in [10], where we use a supervised ranking framework and empirical risk minimization to choose the weights that give the best prediction performance. This approach requires choosing a smooth, preferably convex, loss function for the optimization problem. In [10] we use the exponential loss as a surrogate for area under the ROC curve (AUC), however in the experiments that follow in Section 3 the evaluation metric was mean average precision. Optimizing for AUC in general does not optimize for mean average precision [11], and we found that the exponential loss was a poor surrogate for mean average precision on the Nameling dataset.

2.5 Collaborative filtering baselines

We use two simple collaborative filtering algorithms as baselines in our experimental results in Section 3. For user-based collaborative filtering, we use the cosine similarity between two users in (3) to compute

$$\text{Score}_{\text{UCF}}(b|x_\ell) = \sum_{i=1}^m \mathbb{1}_{[b \in x_i]} \text{sim}(x_\ell, x_i) \tag{7}$$

For item-based collaborative filtering, for any item b we define $\text{Nbhd}(b)$ as the set of observations containing b: $\text{Nbhd}(b) = \{i : b \in x_i\}$. Then, the cosine similarity between two items is defined as before:

$$\text{sim}_{\text{item}}(b, d) = \frac{|\text{Nbhd}(b) \cap \text{Nbhd}(d)|}{\sqrt{|\text{Nbhd}(b)|}\sqrt{|\text{Nbhd}(d)|}}. \tag{8}$$

And the item-based collaborative filtering score of item b is

$$\text{Score}_{\text{ICF}}(b|x_\ell) = \sum_{d \in x_\ell} \text{sim}_{\text{item}}(b, d). \tag{9}$$

In addition to these two baselines, we consider the extremely simple baseline of ranking items by their frequency in the training set. We call this the frequency baseline.

3 Name Recommendations with the Nameling Dataset

We now demonstrate our similarity-weighted adjusted confidence measure on the Nameling dataset released for the ECML PKDD 2013 Discovery Challenge. We also compare the alternative confidence measures and baseline methods from Section 2. A description of the Nameling dataset can be found in [12], and details about the challenge task can be had in the introduction to these workshop proceedings. For the sake of self-containment, we give a brief description here.

3.1 The Nameling Public Dataset

The dataset contains the interactions of users with the Nameling website `http://nameling.net`, a site that allows its users to explore information about names and provides a list of similar names. A user enters a name, and the Nameling system provides a list of similar names. Some of the similar names are given category descriptions, like "English given names," or "Hypocorisms." There are five types of interactions in the dataset: "ENTER_SEARCH," when the user enters a name into the search field; "LINK_SEARCH," when the user clicks on one of the listed similar names to search for it; "LINK_CATEGORY_SEARCH," when the user clicks on a category name to list other names of the same category; "NAME_DETAILS" when the user clicks for more details about a name; and "ADD_FAVORITE" when the user adds a name to his or her list of favorites. The dataset contains 515,848 interactions from 60,922 users.

The data were split into training and test sets by, for users with sufficiently many "ENTER_SEARCH" interactions, setting the last two "ENTER_SEARCH" interactions aside as a test set. Some other considerations were made for duplicate entries - see the introduction to the workshop proceedings for details. The end result was a training set of 443,178 interactions from the 60,922 users, and a test set consisting of the last two "ENTER_SEARCH" names for 13,008 of the users. The task was to use the interactions in the training set to predict the two

names in the test set for each of the test users by producing for each test user a ranked list of recommended names. The evaluation metric was mean average precision of the first 1000 recommendations - see the proceedings introduction for more details.

3.2 Data Pre-processing

We did minimal data pre-processing, to highlight the ability of similarity-weighted adjusted confidence to perform well without carefully crafted features or manual consideration of special cases. We discarded users with no "ENTER_SEARCH" interactions, which left 54,439 users. For each user i, we formed the set of items x_i as "name, interaction type" for all interactions from that user. For example, "Primrose, ENTER_SEARCH" was the feature indicating that the user did an "ENTER_SEARCH" for the name Primrose. The total feature collection Z contained "name, interaction type" for all of the entries in the interaction database. The total number of items in Z was $n = 34,070$. No other data pre-processing was done.

To form rules, we took as left-hand sides a all individual interaction entries: $\mathcal{A} = Z$. We considered as right-hand sides b all valid names to be recommended (among other things, this excluded names that were previously entered by that user - see the proceedings introduction for details on which names were excluded from the test set). An example rule is "Primrose, ENTER_SEARCH $\rightarrow$ Katniss."

3.3 Results

We applied confidence, adjusted confidence, similarity-weighted confidence, and similarity-weighted adjusted confidence to the training set to generate recommendations for the test users. For the adjusted measures, we found the best performance on the test set with $K = 4$ for similarity-weighted adjusted confidence and $K = 10$ for adjusted confidence, as shown in Figure 1. We also applied the user-based collaborative filtering, item-based collaborative filtering, and frequency baselines to generate recommendations. For all of these recommender system approaches, the mean average precision at 1000 on the test set is shown in Table 1.

Similarity-weighted adjusted confidence gave the best performance, and similarity weighting led to a 4.2% increase in performance over (unweighted) adjusted confidence. The adjustment also led to a 9.7% increase in performance from similarity-weighted confidence to similarity-weighted adjusted confidence. User-based collaborative filtering performed well compared to the frequency baseline, but was outperformed by similarity-weighted adjusted confidence by 11.4%. Item-based collaborative filtering performed very poorly.

An advantage of using association rules as opposed to techniques based in regression or matrix factorization is that there is no explicit error minimization problem being solved. This means that association rules generally do not have the same propensity to overfit as algorithms based in empirical risk minimization.

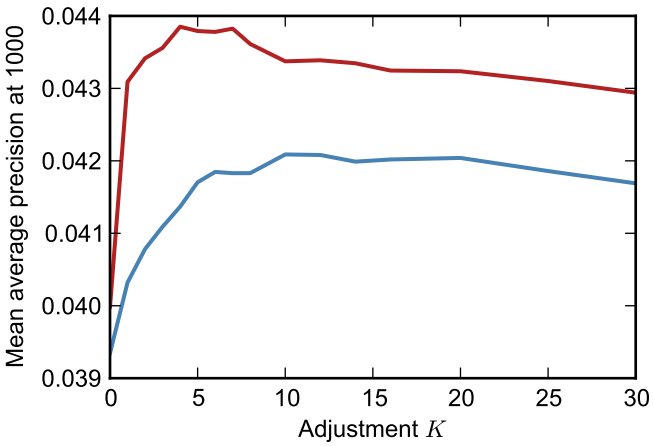

Fig. 1. Test performance for adjusted confidence (blue) and similarity-weighted adjusted confidence (red) for varying amounts of adjustment K.

Table 1. Mean average precision at 1000 for the recommender system approaches discussed in the paper.

Recommender system	Mean average precision
Similarity-weighted adjusted confidence, $K = 4$	0.04385
Adjusted confidence, $K = 10$	0.04208
Similarity-weighted confidence	0.03998
User-based collaborative filtering	0.03936
Confidence	0.03934
Frequency	0.02821
Item-based collaborative filtering	0.01898

We found that the performance on the discovery challenge hold-out dataset was similar to that which we measured on the public test set in Table 1.

4 Conclusions

Similarity-weighted adjusted confidence is a natural fit for the Nameling dataset and the name recommendation task. First, the dataset is extremely sparse (see [12]). The Bayesian adjustment K increases performance by reducing variance for low-support itemsets, and this dataset contains many low-support yet informative itemsets. Second, preferences for names are very heterogeneous. Incorporating the similarity weighting from user-based collaborative filtering into the confidence measure helps to focus the estimation on the more informative users.

Association rules and similarity-weighted adjusted confidence are powerful tools for creating a scalable and interpretable recommender system that will perform well in many domains.

Acknowledgments. Thanks to Stephan Doerfel, Andreas Hotho, Robert Jäschke, Folke Mitzlaff, and Juergen Mueller for organizing the ECML PKDD 2013 Discovery Challenge, and for making their excellent Nameling dataset publicly available. Thanks also to Cynthia Rudin for support and for many discussions on using rules for predictive modeling.

References

1. Zaki, M.J., Ogihara, M.: Theoretical foundations of association rules. In: 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (1998)
2. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work. pp. 241–250. CSCW '00 (2000)
3. McSherry, D.: Explanation in recommender systems. Artificial Intelligence Review 24(2), 179–197 (2005)
4. Herlocker, J.L., Konstan, J.A., Terveen, L.G., John, Riedl, T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22, 5–53 (2004)
5. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. pp. 43–52. UAI'98 (1998)
6. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295. WWW '01 (2001)
7. Rudin, C., Letham, B., Salleb-Aouissi, A., Kogan, E., Madigan, D.: Sequential event prediction with association rules. In: Proceedings of the 24th Annual Conference on Learning Theory. pp. 615–634. COLT '11 (2011)
8. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 226–239 (1998)
9. Tax, D.M., Breukelen, M.V., Duin, R.P., Kittler, J.: Combining multiple classifiers by averaging or by multiplying? Pattern Recognition 33, 1475–1485 (2000)
10. Letham, B., Rudin, C., Madigan, D.: Sequential event prediction. Machine Learning (2013), in press
11. Yue, Y., Finley, T., Radlinski, F., Joachims, T.: A support vector method for optimizing average precision. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 271–278. SIGIR '07 (2007)
12. Mitzlaff, F., Stumme, G.: Recommending given names (2013), `http://arxiv.org/abs/1302.4412`, preprint, arxiv:1302.4412