Distance Metric Learning for Recommender Systems in Complex Domains

First Author Name ( )
Affiliation ( )
Address ( )
e-mail address ( )
Optional phone number ( )

Second Author Name ( )
Affiliation ( )
Address ( )
e-mail address ( )
Optional phone number ( )

ABSTRACT

Recommender systems facilitate reuse, retrieval and exchange of scientific data such as research objects, workflows or experiment plans. The collaborative filtering, content-based filtering and knowledge-based method are the most common techniques for the recommendation task. However, these approaches don't meet the requirements to the recommender systems for scientific items in e-Science. In this paper we propose a novel approach based on the distance metric learning for recommendation of complex objects.

Author Keywords

Reccomender systems; distance metric learning.

ACM Classification Keywords

H.5.m. Novel mechanisms for understanding collaborative patterns and intelligent probing: Miscellaneous

General Terms

Design; Theory.

INTRODUCTION

Recommender systems intend to provide a user with relevant and interesting information according to his personal preferences. Such systems support the user in navigating in a large space of available choices, such as new products or multimedia items. While recommender systems have become a common tool in a wide range of applications - Amazon's product recommendations being on of the most popular examples - they are relatively new to the field of e-Science and e-Research. Studies acknowledge that modern science is increasingly collaborative [9]. However, for a single scientist it is difficult to stay aware of all the work being done that may be relevant to him. Hence, most scientific domains would benefit from the increased efficiency and effectiveness which stems from easily being able to make use of a large amount of collected information. There is clearly a need for a recommender system that facilitate the reuse, retrieval and exchange of research objects such as data, workflows, or experiment plans.

The application of recommender systems in a scientific context is significantly different from the standard case of product recommendations. The biggest issues are the representation of complex objects, a smaller, more heteregeneous set of users and a lack of information about the user's preferences. A recommender system for the field of e-Science must satisfy the following requirements:

1. **Generality** It should be general enough so that it can operate on a wide range of items from different application domains, but specific enough to deliver solid results. The most important is the ability to handle the complex objects that are come from several heterogeneous sources of information.

   As an example let us consider the MyExperiment [13] repository - an environment for collaboration and sharing of workflows and experiment plans. Each workflow stored in MyExperiment is represented by the following components: workflow graph (structured data), Meta information (textual data), statistics about the usage (numerical data) and user information (mixed data). Creation of an appropriate representation that combines all this information is not a trivial task.

2. **Personalized recommendations** As the users base in a scientific collaborative problem is very heterogeneous - the typical usage scenario being small teams of people working on different projects - a single recommendation scheme might not cover the needs of all users. Recommendations need to be more personalized.

   To make the use of the system as easy as possible, the user should not be burdened with the additional trouble of customizing the recommendation by setting up a large set of configurable filters. Instead, the system should use machine learning techniques to adapt itself to the users preferences.

3. **Ability to handle sparse data** In a collaborative context, we are facing with the challenge that while the number of items to recommmend may be as large, or even larger as in a product recommendation context, and while the items are definitively more complex, the number of users might be very low. In particular, as recommendations need to be personalized, it might not be possible to generalize the recommendation function among multiple users, leaving the system only with data from a single user to learn the recommendation function from.

CSCW'12, February 11–15, 2012, Seattle, Washington, USA.

At the same time, a typical user will not be willing to spend a lot of time to set up the recommendation system. Hence, the user should be only asked for input that he can give quickly, and correctly. In particular, it is very favourable to ask the user only questions regarding specific instances, for which domain experts can usually give very concrete feedback, instead of complex, more theoretical questions. As an example, when recommending papers to read, it is better to ask the user "is this paper relevant to you?" instead of "do you like to see more papers from the same author?".

As a consequence, our goal is to set up a recommender system that can learn from complex data with only limited user feedback.

The remainder of the paper is structured as follows: Section 2 gives an overview of existing recommender system. Section 3 explain a distance metric learning approach and its application to the recommendation problem. In Section 4 we investigate the question of how to perform distance metric learning with a few labeling costs. Experimental results are reported in Sections 5. We conclude with Section 6 where we address open issues and future work.

RELATED WORK

Formally, the recommender problem can be described as follows:

Definition:
Let $I = \{i_1, \ldots, i_n\}$ be a set of items and $U = \{u_1, \ldots, u_m\}$ a set of users. Let $Q_u : I \times I \rightarrow R$ be a quality function that measures how close is the predicted ranking of i to the true preferences of the user u regarding a query item $i_q \in I$. Given an user u, we want to learn a recommender function $REC : I \rightarrow I_R, I_R \in \mathcal{P}(I)$ which maps a query item i_q to a set of recommended items I_R having maximal quality. More formaly: $REC(i_q) = \{i \in I \| i = \arg\max_j Q_u(i_j, i_q)\}$

An extensive research towards improving the recommendation quality has been done in the recent years. However, the existing approaches still need further improvements to make recommendations more accurate and to extend the applicability for new type of items [8, 10, 14]. A comprehensive survey of the state-of-the-art in recommender systems is presented by Adomavicius [3].

According to the underlying technique all recommendation approaches can be distinguished into three general categories: collaborative filtering systems, content-based systems and knowledge-based systems.

The most common recommender approach is **collaborative filtering**. It is based on collecting of user's profiles. A typical profile consists of aggregate information about user's preferences that are represented by a set of rated items. The user is recommended items that liked people with similar taste. Some of the most important systems based on this approach are Tapestry [6] and GroupLens [12]. The most common problem of collaboration filtering is the cold-start problem. Since a comparison of the user to the other persons is based solely on his rating information, providing recommendations for new users is not trivial task. To perform well such systems must be initialized with a large amount of collected data. Moreover, the accuracy of such systems is very sensitive in the number of available data. A similar problem is associated with recommendation of a new item.

The **content-based systems** often rely on well known machine learning methods, such as classification or ranking. The **content-based approach** requires the items are represented in some feature space. A new item can be recommended to the user if it is similar to the ones the user liked in the past.

A more detailed overview of content-based filtering systems is presented in [10]. Similar to the collaborative filtering, content-based approaches suffer from start-up problem. To build an accurate prediction model enough data with information about user preferences must be available. The second limitation of this method is the fact that not all items can be easily represented in a feature space. This avoids including some application domains and particularly domains of complex objects that are described by several heterogeneous sources of information.

The **knowledge-based systems** are based either on the knowledge about items or knowledge about users or on both of them. In contrast to the content-based systems, knowledge about items is not restricted to the associated features. This explicit knowledge can be generated automatically by a specific engineering technique or created manually. The knowledge about a user may include demographic characteristics or other information of this kind. In the simplest case it is only defined by the query he has formulated. It is possible to personalized recommendation is enables in two different ways: by several filters that user has to define or through interaction of the system with the user. An example of a knowledge-based system is Entree - a restaurant recommendation system [7] using similarity retrieval.

According to the requirements described above, no one of the existing systems can be directly applied for the recommendation of scientific objects. The knowledge-based systems don't support personalized recommendations (Requirement 1). The context-based approach is not general enough to be applied to the complex object that cannot be represented in a feature space (Requirement 2). Both filtering approaches suffer from the cold-start problem and to perform well require a large amount of data (Requirement 3).

There is obviously a need for a new recommendation technique. The possibility to include explicit knowledge, provided by knowledge-based systems, is a very important feature for e-Science domain. In some cases it is easier to say when two items are similar than to adequately represent them in a feature space. Therefore, we combine a knowledge-based approach with the idea of learning personalized recommendations from user's preferences, which is used in content-based methods. We want to adapt a notion of similarity automatically based on user's preferences using explicit knowledge about items. This problem is called distance metric learning and is presented below.

DISTANCE METRIC LEARNING

Problem Definition

Distance metric learning has received some attention in the field of machine learning [1, 15, 2]. Although this method bears a great potential to solve a wide range problems involving comparison, matching and retrieval, the most state of the art works applied it to improve the classification performance of the KNN algorithm [15] or the quality of the clustering [16].

Distance metric learning differes from the traditional setting of supervised learning in the way the goal is to learn a function that represents the distance between *pairs* of examples. In practice this setting is important because some types of information can be represented more adequately. Having knowledge about the data we can easy define a set of meaningful basic distances and represent each instance pair by a numerical vector. Even if the original instance cannot be represented in some feature space.

Let $i_1, \ldots, i_n \in I$ be a set of instances and $D = \{d_1, \ldots, d_k\}, d_i : I \times I \to [0, 1]$ a set of distance functions called local distances. We know about some instance pairs whether they are similar/dissimilar: $(i, i') \in S$ or $(i, i') \in D$, also called equivalence/inequivalence constraints. The goal is to find a function $dist = f(d_1, \ldots, d_k) : I \times I \to [0, 1]$ that is defined solely on the values d_i and that optimally reflects the similarity given by the training data. The general problem of the metric learning can be formulated as an optimization of a cost function. The way of solving this problem can be different depending on the specific type of the cost function and the assumptions that are made for data distribution.

Xing et. all [16] formulate the distance metric learning as a constrained convex optimisation problem. The Euclidean distance between two instances $x, y \in R^n$ is parameterized by a positive semidefinite matrix A is defined as:

Recommender Function using Distance Metric

The recommender function aims to provide the user by a set of the most similar items regarding the query item i_q. More formally:

Definition:
Let $I = \{i_1, \ldots, i_n\}$ be a set of items and $d : I \times I \to R$ - a distance function learned from a set of labeled pairs. Given a query item i_q, the recommendation function $REC : I \to \mathcal{I}$ deliver a set of k items that are most similar to the i_q. More formally, $REC(i_q) = \{i \in I \| i = \arg\max_j d(i_j, i_q)\}$

Thus, we can easily apply a distance metric for recommendation tasks. Distance metric learning does not requires that the instances are represented in some feature space. It only assumes that meaningfyll local distances can be defined. adequately Thus, it serves a good solution for a wide range of different items, including complex objects. It is also one of the most promising techniques in cases where the notion of the similarity is user-dependent. It enables to learn a user-specific similarity function. The weack point of this approch is a need for training data. We address this problem in the next Section, where we investigate the question of how to learn a distance metric from a few labeled data. Experimental results showed that appropriately-designed sampling strategy can significantly reduce user efforts in creation of personalized similarity function.

LEARNING WITH FEW LABELING COSTS

In order to learn a distance metric according to user's preferences the user has to give feedback about the similarity of some item pairs. It becomes obvious that the number of pairs that are selected to be shown to the user should be as small as possible. The idea is to select a small set of pairs that is informative enough to create a good model. Sampling is the common approach to this problem.

To formulate sampling task in the setting of distance metric learning first consider the input data. Given a set of n items $i_1, \ldots, i_n \in I$ and a set of distance functions $d_1, \ldots, d_k : X \times X \to [0, 1]$ defined by the domain expert. We transform the original items into the distance space over item pairs $\mathcal{P}$, which is defined as follow: $\forall i, i' \in X$: $p_i(i, i') = (d_1(i,'), \ldots d_k(i, i')) \in \mathbb{R}$.

Now the sampling problem can be formulated:

Definition:
Given n items, find an optimal distance metric under the condition that a user have to label at most k instance pairs, where $k \ll n$.

What we need is an intelligent sampling strategy that selects the most 'interesting' pairs from a pool of unlabeled data to show them to the user. Next, we present four approaches aimed to solve this task.

1. **Random Sampling** We randomly sample k independent instance pairs from the available ones.

2. **Euclidean Sampling** The idea is to select the most similar and the most dissimilar pairs according to Euclidean distance between instances. So to get k pairs for training we select $m = k/2$ pairs with the largest distance and the $k - m$ pairs with the shortest distance.

3. **Simplex Volumen Maximization Approach (SIVM)** We exploit the idea of describing massive data sets by using latent components proposed by Thurau et. al [4]. We adopted the algorithm presented in the paper for sampling the most 'informative' instance pairs.

   The main idea is to describe data by representing it as a linear combination of basic vectors that usually correspond to the most extreme data points. Moreover, these vectors span a simplex that encloses most of the remaining data. More details about the algorithm can be found in [4].

4. **Simplex Approach** The idea of a simplex method is similar to the SIVM approach. To create a meaningful training data we select the instance pairs that are the most extreme points according to the local distance d_i. More formally: $p_i \in P_s : \arg\max_{p \in \mathcal{P}} < p, u_i >$, where u_i is a unit vector codirectional with the i-th axe of distance space.

EXPERIMENTAL RESULTS

In this section we show that the distance metric learning enables to achieve a good results by comparison of complex objects. Further, we investigate the question whether it is possible to reduce labeling costs in distance metric learning.

1. **Distance Metric Learning for Comparison of Complex Objects** In our previous work [11] we considered the question of how to recommend an appropriate algorithm for a dataset at hand. We reduce this task to the comparison of datasets, which can be seen as complex objects. Since a description of a dataset includes heterogeneous types of information - Meta attributes, numerical values, structured information - it is hard to compare by standard methods. We showed that similarity function learned by distance metric approach significantly improve the quality of recommendation.

2. **Minimizing labeling costs in distance metric learning** We investigate whether an appropriately-designed sampling strategy enables to reduce a number of training pairs that are needed to learn similarity according to user's preferences.

   We have evaluated four sampling strategies regarding their ability to produce an accurate distance metric. The quality of the learned distance metric was evaluated on a supervised classification task. The evaluation was performed on a set of 9 UCI [5] datasets. The results showed that application of intelligent sampling strategies enables to produce an accurate distance metric with few labeling costs.

The experimental results show that the distance metric learning can be successfully applied for comparison of complex objects, such as datasets. Using an intelligent sampling strategy enables to learn an accurate distance metric on few training data, which is relevant for the sparse labeled data in e-Science.

CONCLUSION

The recommender systems aim to support a user in making a decision about a large amount of available choices. Providing recommendation about scientific data, such as datasets, workflows or experiment plans facilitate their reuse and exchange in collaborative research.

In our work we formulate the requirements to the recommender systems that can be applied in the field of e-Science. Then we propose a novel approach that combines the ideas of the knowledge-based method and the content-based filtering. Finally we show that an appropriately-designed sampling strategy enables to avoid the cold-start problem and helps to provide the user by personalized recommendations.

REFERENCES

1. Adam, W., and Alexandros, K. A new framework for dissimilarity and similarity learning. In *PAKDD (2)*, Lecture Notes in Computer Science, Springer (2010), 386–397.
2. Adam, W., Alexandros, K., and Melanie, H. Learning to combine distances for complex representations. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, ACM (2007), 1031–1038.
3. Adomavicius, G., and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering 17*, 6 (2005), 734–749.
4. Christian, T., Kristian, K., Wahabzada, M., and Bauckhage, C. Descriptive matrix factorization for sustainability: Adopting the principle of opposites. *Journal of Data Mining and Knowledge Discovery* (2011).
5. Frank, A., and Asuncion. UCI machine learning repository, 2010.
6. Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM 35* (December 1992), 61–70.
7. Kolodner, J. Using collaborative filtering to weave an information tapestry. *San Mateo, CA: Morgan Kaufmann* (1993).
8. Mobasher, B. Data Mining for Web Personalization. In *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer (2007), 90–135.
9. Olson, G. M., Zimmerman, A., and Bos, N. *Scientific Collaboration on the Internet*. The MIT Press, 2008.
10. Pazzani, M., and Billsus, D. Content-Based Recommendation Systems The Adaptive Web. In *The Adaptive Web*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, 325–341.
11. Punko, N., Rueping, S., and Stefan, W. Facilitating clinico-genomic knowledge discovery by automatic selection of kdd processes, July 2008.
12. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94 (1994), 175–186.
13. Roure, D. D., Goble, C., Bhagat, J., Cruickshank, D., Goderis, A., Michaelides, D., and Newman, D. myexperiment: Defining the social virtual research environment. In *4th IEEE International Conference on e-Science*, IEEE Press (December 2008), 182–189.
14. Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. The adaptive web. Springer-Verlag, Berlin, Heidelberg, 2007, ch. Collaborative filtering recommender systems, 291–324.
15. Weinberger, K. Q., Blitzer, J., and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, MIT Press (2006).
16. Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, MIT Press (2002), 505–512.