Semi-Supervised Learning by Mixed Label Propagation

Wei Tong and **Rong Jin**
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
{tongwei, rongjin}@cse.msu.edu

Abstract

Recent studies have shown that graph-based approaches are effective for semi-supervised learning. The key idea behind many graph-based approaches is to enforce the consistency between the class assignment of unlabeled examples and the pairwise similarity between examples. One major limitation with most graph-based approaches is that they are unable to explore *dissimilarity* or negative similarity. This is because the dissimilar relation is not transitive, and therefore is difficult to be propagated. Furthermore, negative similarity could result in unbounded energy functions, which makes most graph-based algorithms unapplicable. In this paper, we propose a new graph-based approach, termed as "**mixed label propagation**" which is able to effectively explore both similarity and dissimilarity simultaneously. In particular, the new framework determines the assignment of class labels by (1) minimizing the energy function associated with positive similarity, and (2) maximizing the energy function associated with negative similarity. Our empirical study with collaborative filtering shows promising performance of the proposed approach.

Introduction

Recent studies have shown promising performance of graph-based approaches for semi-supervised learning (Altun, McAllester, & Belkin 2005; Belkin, Niyogi, & Sindhwani 2006; Chu & Ghahramani 2005; Herbster, Pontil, & Wainer 2005; Joachims 2003; Zhou, Schölkopf, & Hofmann 2005; Zhou *et al.* 2004; Zhu, Ghahramani, & Lafferty 2003). The key idea behind most graph-based approaches is to explore the pairwise similarity between examples in determining the class labels for unlabeled examples. In particular, the class assignments of unlabeled examples need to be consistent with both the example similarity and the class labels of training examples. Graph-based approaches can often be interpreted as propagating the label information of the training examples to the unlabeled examples through the pairwise similarity between examples. This process is sometimes referred to as label propagation (Zhou *et al.* 2004).

Despite the success, most graph-based approaches are limited to explore positive similarity, which can be inter-

preted as the confidence of assigning two different examples to the same class. In many cases, we may run into dissimilarity, or negative similarity, that expresses the confidence of assigning two examples to different classes. For instance, if we measure the similarity between two examples by their correlation coefficient, we could have both positive and negative similarity. One application of negative similarity is collaborative filtering, in which a negative similarity between two users indicates that the two users share different interests and therefore tend to give opposite ratings for the same items. Another application of negative similarity is semi-supervised data clustering, in which one has side information of must link pairs and must not link pairs. One way to explore the side information is to associate every must link pair with a positive similarity, and every must not link pair with a negative similarity (Kulis *et al.* 2005).

It is important to note that most existing graph-based approaches are unapplicable to negative similarity because the dissimilar relations are non-transitive, and therefore can not be propagated directly. This can also be understood from the viewpoint of optimization. The energy function, i.e., the objective function employed by most graph-based approaches to measure the inconsistency between the class assignments and the example similarity, could be negatively unbounded when similarity is negative, thus, no optimal solution can be found to minimize the objective function. To address this problem, we propose a new framework of label propagation for semi-supervised learning, termed as **mixed label propagation** which can effectively explore both negative and positive similarity simultaneously. It minimizes the *inconsistency* between the class assignments and positive similarity, and in the meantime maximizes the *consistency* between the class assignments and negative similarity. Our empirical study with collaborative filtering shows that the proposed approach is effective in exploring the negative similarity. It is worth pointing out that a highly related paper (Goldberg, Zhu, & Wright 2007) was published just after the submission of this paper, in which the authors incorporated both similarity and dissimilarity by introducing an auxiliary matrix W, where $W_{ij} = -1$ if the similarity between the two samples is negative, otherwise $W_{ij} = 1$. Then, they used the $L + (\mathbf{1} - W) \bullet S$ to replace the Laplacian matrix L in the energy function, where $\mathbf{1}$ is the all-one matrix, S is the similarity matrix and $\bullet$ is the elementwise product.

The rest of the paper is arranged as follows: we first review the related work on graph-based approaches and collaborative filtering, which is used in our empirical evaluation. Then, we describe the framework of mixed label propagation and its application to collaborative filtering. The results of our empirical studies are presented in the experiment section and in the last section we conclude our work.

Related Work

We first review the previous work on graph-based approaches for semi-supervised learning, followed by a brief overview of collaborative filtering, which is used in our empirical study for evaluating the proposed approach.

Graph-based Approaches   The main idea of graph-based approaches is to search for the class assignments of the unlabeled examples that are consistent with the pairwise similarity between any two examples. Many graph-based approaches have been developed in the past, including the harmonic approach (Zhu, Ghahramani, & Lafferty 2003), the Green's function approaches (Zhou *et al.* 2004; Zhou, Schölkopf, & Hofmann 2005), spectral graph transducer (Joachims 2003), the online approach (Herbster, Pontil, & Wainer 2005), and the Gaussian process (Altun, McAllester, & Belkin 2005). One key component to most graph-based approaches is how to measure the inconsistency between the class assignments and the example similarity. For instance, in the harmonic function approach, the inconsistency between the class assignment $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ and similarity $S_{ij} \geq 0$ is measured by the energy function:

$$E(S, \mathbf{y}) = \sum_{i,j=1}^{n} S_{i,j} (y_i - y_j)^2 = \mathbf{y}^\top L \mathbf{y} \qquad (1)$$

where L is the Laplacian matrix and is defined as $L = D - S$. Here, $D = \text{diag}(D_1, D_2, \ldots, D_n)$ is a diagonal matrix with its diagonal elements defined as $D_i = \sum_{j=1}^{n} S_{i,j}$. Given the class labels $\hat{\mathbf{y}}_l = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{n_l})$ for the first n_l examples, the optimal class assignment $\mathbf{y}$ is found by minimizing the above energy function, i.e.,

$$\min_{\mathbf{y}} \quad E(S, \mathbf{y}) \qquad (2)$$
$$\text{s. t.} \quad \mathbf{y}_l = \hat{\mathbf{y}}_l$$

where $\mathbf{y}_l$ stands for the first n_l elements of $\mathbf{y}$. The optimal class labels assigned to the unlabeled examples, denoted by $\mathbf{y}_u$, are computed as

$$\mathbf{y}_u = -[L^{u,u}]^{-1} L^{u,l} \hat{\mathbf{y}}_l \qquad (3)$$

where the super indices u and l stand for the parts of the Lapacian matrix that are related to the labeled and the unlabeled examples, respectively.

It is important to note that the pairwise similarity $S_{i,j}$ in the above energy function must be non-negative. This is because $E(S, \mathbf{y})$ could become negatively unbounded when certain pairwise similarity $S_{i,j}$ is negative, which implies the optimal solution to (3) does not exist. The proposed approach addresses this problem by measuring the two quantities: the inconsistency between the class assignments and

the positive similarity, and the consistency between the class assignments and the negative similarity. The optimal class assignments are found by minimizing the ratio between the inconsistency and the consistency.

Collaborative Filtering   The goal of collaborative filtering is to predict the utility of items to a user based on the ratings by other users (Resnick *et al.* 1994). To predict the ratings of items by an user u, the key idea behind most collaborative filtering algorithms is to first identify a subset of users who share similar interests to u, and then combine the ratings of these similar users as the ratings by u. The most well known algorithms for collaborative filtering include Pearson correlation coefficient (Resnick *et al.* 1994), personality diagnosis (Pennock *et al.* 2000), matrix factorization (Srebro, Rennie, & Jaakkola 2005), graphical models (Breese, Heckerman, & Kadie 1998; Hofmann 2003), and ordinal regression (Chu & Ghahramani 2005).

Mixed Label Propagation

In this section, we will first present the general framework of mixed label propagation for semi-supervised learning, followed by the description of an efficient algorithm and the application to collaborative filtering.

The Framework of Mixed Label Propagation

To incorporate negative similarity into the framework of label propagation, we consider constructing two energy functions: the energy function E_+ that measures the *inconsistency* between the class assignments and the positive similarity, and the energy function E_- that measures the *consistency* between the class assignments and the negative similarity. In order to minimize the inconsistency E_+ and maximize the consistency E_- simultaneously, we follow the idea of Linear Discriminative Analysis (LDA) (Fisher 1936) by minimizing the ratio between E_+ and E_-. More specifically, given the pairwise similarity S, we construct the positive similarity matrix S_+ and the negative similarity matrix S_- as follows

$$[S_+]_{i,j} = \begin{cases} S_{i,j} & S_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$[S_-]_{i,j} = \begin{cases} |S_{i,j}| & S_{i,j} < 0 \\ 0 & \text{otherwise} \end{cases}$$

Evidently, we have $S = S_+ - S_-$. We then construct two energy functions $E_+(S_+, \mathbf{y})$ and $E_-(S_-, \mathbf{y})$ based on the two similarity matrices S_+ and S_- using Eqn. (1). Finally, given the class labels $\hat{\mathbf{y}}_l \in \{-1, +1\}^{n_l}$ for the first n_l training examples, the optimal class assignment $\mathbf{y}$ is determined by minimizing the ratio between E_+ and E_-, i.e.,

$$\min_{\mathbf{y} \in \mathbb{R}^n} \quad \frac{E_+(S_+, \mathbf{y})}{E_-(S_-, \mathbf{y})} = \frac{\mathbf{y}^\top L_+ \mathbf{y}}{\mathbf{y}^\top L_- \mathbf{y}} \qquad (4)$$
$$\text{s. t.} \quad y_i = \hat{y}_i, \ i = 1, 2, \ldots, n_l$$

where L_+ and L_- are graph Laplacians for similarity matrices S_+ and S_-, respectively. Note that, without the linear constraints $y_i = \hat{y}_i, i = 1, 2, \ldots, n$, the above optimization problem is identical to the optimization problem

in LDA (Fisher 1936). Hence, the optimal solution $\mathbf{y}$ to (5) is the minimum eigenvector of matrix $L_-^\dagger L_+$ where $\dagger$ stands for the pseudo inverse. The challenges of solving the optimization problem in (5) arises from the linear constraints. In next subsection, we present an efficient algorithm to solve this optimization problem.

An Efficient Algorithm for Mixed Label Propagation

For the convenience of presentation, we rewrite class assignments $\mathbf{y}$ as $\mathbf{y} = (\mathbf{y}_l, \mathbf{y}_u)$, where $\mathbf{y}_l$ represents the class labels of the first n_l examples and $\mathbf{y}_u$ represents the labels for the next $n_u = n - n_l$ examples. According to the constraints in (5), we have $\mathbf{y}_l = \hat{\mathbf{y}}_l$.

To solve the problem in (5), we first follow the idea of LDA by converting the problem of optimizing a ratio into a constrained optimization problem, i.e.,

$$\min_{\beta \in \mathbb{R}, \mathbf{y}_u \in \mathbb{R}^{n_u}} \quad \mathbf{y}^\top L_+ \mathbf{y} \tag{5}$$
$$\text{s. t.} \quad \mathbf{y}^\top L_- \mathbf{y} \geq 1, \quad \beta \geq 0$$

where

$$\mathbf{y}^\top L_+ \mathbf{y} = (\beta \hat{\mathbf{y}}_l^\top, \mathbf{y}_u^\top) \begin{pmatrix} L_+^{l,l} & L_+^{l,u} \\ L_+^{u,l} & L_+^{u,u} \end{pmatrix} \begin{pmatrix} \beta \hat{\mathbf{y}}_l \\ \mathbf{y}_u \end{pmatrix}$$

$$\mathbf{y}^\top L_- \mathbf{y} = (\beta \hat{\mathbf{y}}_l^\top, \mathbf{y}_u^\top) \begin{pmatrix} L_-^{l,l} & L_-^{l,u} \\ L_-^{u,l} & L_-^{u,u} \end{pmatrix} \begin{pmatrix} \beta \hat{\mathbf{y}}_l \\ \mathbf{y}_u \end{pmatrix}$$

Note that, in (5), we introduce a scaling factor β for $\hat{\mathbf{y}}_l$. This is because we introduce the constraint $\mathbf{y}^\top L_- \mathbf{y} \geq 1$, and therefore have to convert $\mathbf{y}_l = \hat{\mathbf{y}}_l$ to $\mathbf{y}_l \propto \hat{\mathbf{y}}_l$. β is introduced to account for the scaling factor between $\mathbf{y}_l$ and $\hat{\mathbf{y}}_l$.

We take the alternative optimization strategy to solve (5). More specifically, we first optimize $\mathbf{y}_u$ by fixing β, and then optimize β by fixing $\mathbf{y}_u$. However, the problem in (5) is a non-convex programming problem for both β and $\mathbf{y}_u$ because of the non-convex constraint $\mathbf{y}^\top L_- \mathbf{y} \geq 1$. To resolve this problem, we resort to the following theorem of the alternative (Boyd & Vandenberghe 2004):

Theorem 1. *The implication*

$$\mathbf{x}^\top F_1 \mathbf{x} + 2g_1^\top \mathbf{x} + h_1 \leq 0 \Longrightarrow \mathbf{x}^\top F_2 \mathbf{x} + 2g_2^\top \mathbf{x} + h2 \leq 0,$$

where F_i is symmetric $n \times n$ matrix, holds if and only if there exists $\lambda \geq 0$ such that

$$\begin{pmatrix} F_2 & g_2 \\ g_2^\top & h_2 \end{pmatrix} \succeq \lambda \begin{pmatrix} F_1 & g_1 \\ g_1^\top & h_1 \end{pmatrix}$$

Optimize y with fixed β Using the above theorem, to compute the optimal $\mathbf{y}_u$ with fixed β, we turn the problem in (5) into its dual form, i.e.,

$$\max_{\lambda} \quad \lambda a_{22} - \beta^2 a_{21} a_{11}^{-1} a_{12} \tag{6}$$
$$\text{s. t.} \quad \lambda \geq 0$$
$$a_{11} = L_+^{u,u} - \lambda L_-^{u,u}, a_{12} = \left(L_+^{u,l} - \lambda L_-^{u,l}\right) \hat{\mathbf{y}}_l$$
$$a_{21} = \hat{\mathbf{y}}_l^\top \left(L_+^{l,u} - \lambda L_-^{l,u}\right), a_{22} = 1 - \beta^2 \hat{\mathbf{y}}_l^\top L_-^{l,l} \hat{\mathbf{y}}_l$$

Given the solution for λ, we can compute the solution for $\mathbf{y}_u$ using the Karush-Kuhn-Tucker (KKT) conditions (Boyd & Vandenberghe 2004), i.e.,

$$\mathbf{y}_u = -\beta \left(L_+^{u,u} - \lambda L_-^{u,u}\right)^{-1} \left(L_+^{u,l} - \lambda L_-^{u,l}\right) \mathbf{y}_l \tag{7}$$

It is interesting to note that the above solution for $\mathbf{y}_u$ is equivalent to the solution by the harmonic function (in Eqn. (3)) if we use $L = L_+ - \lambda L_-$ as the graph Laplacian matrix. Thus, the parameter λ weights the importance between the two energy functions E_+ and E_-. The advantage of the proposed approach is that it automatically determines λ by making the optimal tradeoff between the inconsistency measure E_+ and the consistency measure E_-. This is particularly important for semi-supervised learning when the number of labeled examples is limited and is insufficient to determine λ by cross validation. We will also show in our empirical study that the value of λ varies significantly from one case to another, and therefore it is suboptimal to replace λ with a fixed constant. The problem in (6) can be further turned into a Semi-Definite Programming (SDP) problem as follows:

$$\max_{\lambda, \gamma} \quad \gamma \tag{8}$$
$$\text{s. t.} \quad \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & (\lambda a_{22} - \gamma)/\beta^2 \end{pmatrix} \succeq 0, \ \lambda \geq 0$$
$$a_{11} = L_+^{u,u} - \lambda L_-^{u,u}, \ a_{12} = \left(L_+^{u,l} - \lambda L_-^{u,l}\right) \hat{\mathbf{y}}_l$$
$$a_{21} = \hat{\mathbf{y}}_l^\top \left(L_+^{l,u} - \lambda L_-^{l,u}\right), a_{22} = 1 - \beta^2 \hat{\mathbf{y}}_l^\top L_-^{l,l} \hat{\mathbf{y}}_l$$

In (8), we introduce the slack variable $\gamma \leq \lambda a_{22} - \beta^2 a_{21} a_{11}^{-1} a_{12}$, which can be further turned into a Linear Matrix Inequality (LMI)

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & (\lambda a_{22} - \gamma)/\beta^2 \end{pmatrix} \succeq 0$$

using the Schur complement (Boyd & Vandenberghe 2004). Since the problem in (8) belongs to the family of semi-definitive programming, it can be solved effectively using the standard packages such as SeDuMi[1].

Optimize β with fixed y Similarly, using the theorem 1, we have the following optimization problem for finding optimal β with fixed $\mathbf{y}_u$

$$\max_{\lambda, \gamma} \quad \gamma \tag{9}$$
$$\text{s. t.} \quad \lambda \geq 0, \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \succeq 0$$
$$b_{11} = \hat{\mathbf{y}}_l^\top L_+^{l,l} \hat{\mathbf{y}}_l - \lambda \hat{\mathbf{y}}_l^\top L_-^{l,l} \hat{\mathbf{y}}_l$$
$$b_{12} = b_{21} = \hat{\mathbf{y}}_l^\top L_+^{l,u} \mathbf{y}_u - \lambda \hat{\mathbf{y}}_l^\top L_-^{l,u} \mathbf{y}_u$$
$$b_{22} = -\lambda \left(\mathbf{y}_u^\top L_-^{u,u} \mathbf{y}_u - 1\right) - \gamma$$

The corresponding solution for β is

$$\beta = -\frac{\hat{\mathbf{y}}_l^\top L_+^{l,u} \mathbf{y}_u - \lambda \hat{\mathbf{y}}_l^\top L_-^{l,u} \mathbf{y}_u}{\hat{\mathbf{y}}_l^\top L_+^{l,l} \hat{\mathbf{y}}_l - \lambda \hat{\mathbf{y}}_l^\top L_-^{l,l} \hat{\mathbf{y}}_l} \tag{10}$$

---

[1] http://sedumi.mcmaster.ca/

	Given 10 Rated Items	
	$K_c = 1$	$K_c = 5$
MLP	**0.8318** ± 2.7E-04	**0.7368** ± 1.2E-04
LP	0.8184 ± 8.6E-04	0.7316 ± 1.3E-04
Pearson	0.7766 ± 6.8E-04	0.6749 ± 4.0E-04
MMMF	0.7827 ± 9.2E-05	0.6974 ± 3.6E-05
	Given 15 Rated Items	
	$K_c = 1$	$K_c = 5$
MLP	**0.8599** ± 3.4E-04	**0.7704** ± 1.2E-04
LP	0.8526 ± 5.3E-04	0.7689 ± 1.4E-04
Pearson	0.8170 ± 1.0E-03	0.7222 ± 3.8E-04
MMMF	0.8189 ± 2.0E-04	0.7221 ± 8.4E-05

Table 1: Average precision for the Mixed Label Propagation (MLP), Label Propagation (LP), Pearson Correlation Coefficient (Pearson) and Maximum-Margin Matrix Factorization(MMMF) using 10 training users.

	Given 10 Rated Items	
	$K_c = 1$	$K_c = 5$
MLP	**0.8325** ± 3.7E-04	**0.7449** ± 1.3E-04
LP	0.8228 ± 1.3E-04	0.7408 ± 4.7E-05
Pearson	0.7998 ± 3.5E-04	0.6938 ± 3.4E-04
MMMF	0.7946 ± 1.5E-04	0.7036 ± 1.2E-04
	Given 15 Rated Items	
	$K_c = 1$	$K_c = 5$
MLP	**0.8661** ± 8.8E-05	**0.7732** ± 5.2E-05
LP	0.8592 ± 4.7E-04	0.7727 ± 1.4E-04
Pearson	0.8177 ± 3.3E-04	0.7291 ± 1.9E-04
MMMF	0.8128 ± 2.3E-05	0.7219 ± 1.5E-04

Table 2: Average precision for the Mixed Label Propagation (MLP), Label Propagation (LP), Pearson Correlation Coefficient (Pearson) and Maximum-Margin Matrix Factorization(MMMF) using 20 training users.

In summary, we start with a large value for β, then find the optimal $\mathbf{y}_u$ by solving the problem in (8) with fixed β, and find the optimal β by solving the problem in (9) with fixed $\mathbf{y}_u$. We alternate these two steps iteratively until the solution converges to the local maximum.

Application to Collaborative Filtering

In order to apply the proposed approach to collaborative filtering, the key is to estimate the matrix S for user similarity. To this end, we employ the Pearson correlation coefficient (Resnick *et al.* 1994) to estimate user similarity. It measures the linear correlation between the ratings of two users. More specifically, given two users u_i and u_j, let $\mathcal{O}^{i,j}$ denote the set of items that are rated by both users. We then measure the similarity between u_i and u_j as

$$S_{i,j} = \frac{\sum_{k=1}^{|\mathcal{O}^{i,j}|}(r_i(k) - \bar{r}_i)(r_j(k) - \bar{r}_j)}{\sqrt{\sum_{k=1}^{|\mathcal{O}^{i,j}|}(r_i(k) - \bar{r}_i)^2}\sqrt{\sum_{k=1}^{|\mathcal{O}^{i,j}|}(r_j(k) - \bar{r}_j)^2}}$$

where $r_i(k)$ stands for the rating of the kth item by user u_i, and $\bar{r}_i$ is the average rating of user u_i. Since Pearson correlation coefficient can be both negative and positive, we can apply the proposed method to collaborative filtering.

Experiments

We evaluated the effectiveness of the proposed mixed label propagation approach by the task of collaborative filtering. We used a subset of MovieLens database (http://movielens.umn.edu/login) as the test bed. In particular, we selected the 100 most popular movies, and randomly selected 200 users who had provided at least 25 ratings for the 100 selected movies. In contrast to the binary label requirement in (5), the labels here were the ratings ranging from 1 to 5, and the mixed label propagation algorithm was applied to each movie separately to predict the ratings by all users. To acquire a full spectrum of the performance, we varied the number of training users and the

number of movies whose ratings were provided by the test users. More specifically, 10 and 20 users were used as the training users. For each test user, 10 and 15 movies were randomly selected, and their ratings by the test user were given. Each experiment was conducted ten times, and the results averaged over ten trials were reported in our study.

Three baseline models were used in our study: the Pearson correlation coefficient method (Pearson) (Resnick *et al.* 1994), the Maximum-Margin Matrix Factorization method (MMMF) (Rennie & Srebro 2005) and the Label Propagation method (LP) (Zhu, Ghahramani, & Lafferty 2003) that is based on the harmonic function and only uses the positive similarity. By comparing to the Pearson correlation method and the maximum-margin matrix factorization method, we are able to observe if the idea of proposed method is effective for collaborative filtering. By comparing to the label propagation method, we are able to observe if the proposed method is effective exploiting negative similarity. For maximum margin matrix factorization method, we used the code from the website http://people.csail.mit.edu/nati/mmmf/code.html and used maximum norm with $C = 0.001$ for all the experiments following the suggestion in (Srebro, Rennie, & Jaakkola 2005).

To examine the quality of different collaborative filtering algorithms, we focused on evaluating how well each method was able to rank items for users. For each user, we ranked the items in the descending order of their estimated ratings. We then evaluated the ranked items by comparing to the items ranked by the true ratings. More specifically, for a test user u, we used $\mathbf{l}_e = (i_1, i_2, \ldots, i_m)$ to denote the list of items ordered by the estimated ratings, and r_i^u to denote the true rating of the ith item by user u. Then, the quality of the ranked items was evaluated by the following two metrics:

- *Average Precision* (Freund *et al.* 1998) (AP). To measure the average precision, we assume that the first k items with the highest ratings by user u, denoted by $\mathcal{M}_u(k)$, are the "good" items for user u. Then, the average precision

for user u at the cutoff rank K_c is computed as:

$$AP_u(K_c) = \frac{1}{K_c} \sum_{k=1}^{K_c} \frac{|\mathcal{M}_u(k) \cap \{i_1, \cdots, i_k\}|}{k} \qquad (11)$$

where $|\cdot|$ outputs the length of the set.

- *Average Rating* (AR). It computes the average rating of the first K_r ranked items in the list $\mathbf{l}_e$. More specifically, the average rating for user u at the rank K_r is computed as follows

$$AR_u(K_r) = \frac{1}{K_r} \sum_{k=1}^{K_r} r_{i_k}^u \qquad (12)$$

Finally, we averaged both metrics over all test users, and reported the average precision at different cutoff rank K_c and average rating at different ranking position K_r. Note that we did not use the Mean Average Error (MAE) (Breese, Heckerman, & Kadie 1998) because MAE requires an additional step to calibrate scores into rating, and therefore does not directly reflect the quality of collaborative filtering algorithms.

## Experiment (I): Effectiveness of Mixed Label Propagation

The average precisions of the four methods for 10 and 20 training users are reported in Table 1 and 2, respectively. Figure 1 and 2 show the average ratings of the four methods for 10 and 20 training users, respectively. First, we observe that for all of the four methods, both the average precision and the average rating are improved with larger number of training users and more given rated items. This is consistent with our expectation: the more the training examples, the better the performance. Second, according to the average precision metric, we observe that compared to other methods, the Pearson correlation coefficient method achieves almost the worst performance. It is also surprising to observe that the maximum margin matrix factorization (i.e., MMMF) does not improve the prediction accuracy in comparison to the Pearson correlation coefficient. We believe that this may be attributed to the small number of training users used in our study while most of the previous studies of MMMF focused on large numbers of training users. Third, the label propagation method based on the harmonic function performs better than both Pearson correlation coefficient method and the maximum-margin matrix factorization method according to the average precision metric. Finally, according to both metrics, the mixed label propagation method outperforms the other three methods considerably in all experiments.

Experiment (II): Empirical Values for λ

As described before, the solution for $\mathbf{y}_u$ in Eqn. (7) is essentially equivalent to the solution by the harmonic function (in Eqn. (3)) if we use $L = L_+ - \lambda L_-$ as the graph Laplacian matrix. Since we have to solve the mixed label propagation problem for each movie, we compute a different λ for each movie. Figure 3 shows the λs that were computed for the 100 movies with 20 training users and 10 given ratings. We clearly see the large variance in λ across different movies.
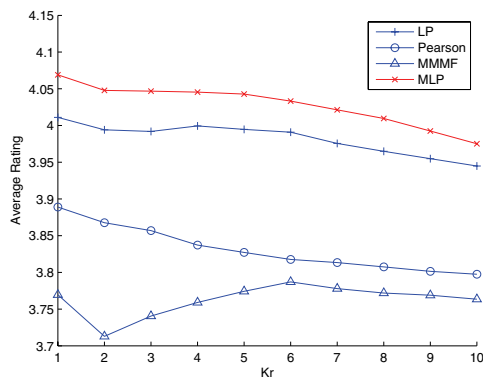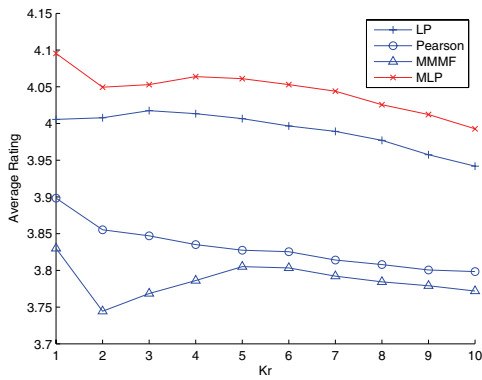

(a) 10 given rated items
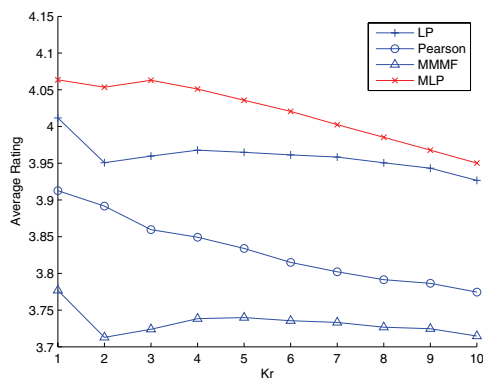
(b) 15 given rated items

Figure 1: The Average rating of the Mixed Label Propagation (MLP), Label Propagation (LP), Pearson Correlation Coefficient (Pearson) and Maximum-Margin Matrix Factorization(MMMF) using 10 training users.

For some movies, the optimal λ can be as high as 0.3. For other movies, the optimal λ can be as low as 10^{-4}. Given the empirical values of λ shown in Figure 3, it is unlikely to find a fixed λ that can fit in well with all movies. This is also confirmed by our empirical study with fixed λ.

Conclusions

In this paper, we proposed a mixed label propagation framework for semi-supervised learning. Unlike the existing graph-based approaches that are only applicable to the positive similarity of examples, our framework is able to explore both positive and negative similarity simultaneously. The key idea behind the proposed framework is to minimize the inconsistency between the class assignments and the positive similarity of examples, and maximize the consistency between the class assignments and the negative similarity of examples. We presented an efficient learning algorithm for the mixed label propagation that is based on the alternative optimization strategy and semi-definitive programming. Our empirical study with collaborative filtering showed that the proposed algorithm is effective in exploring negative simi-

(a) 10 given rated items
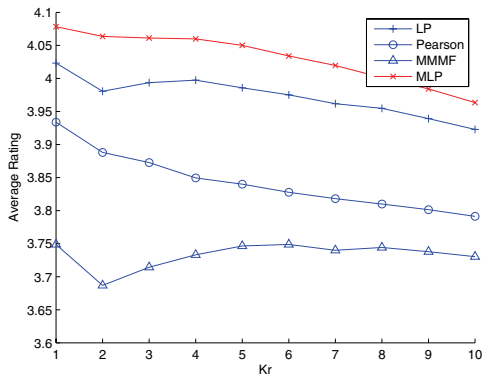
(b) 15 given rated items

Figure 2: The Average rating of the Mixed Label Propagation (MLP), Label Propagation (LP), Pearson Correlation Coefficient (Pearson) and Maximum-Margin Matrix Factorization(MMMF) using 20 training users.

larity and outperforms both the label propagation approach and state-of-the-art approaches for collaborative filtering.

References

Altun, Y.; McAllester, D. A.; and Belkin, M. 2005. Margin semi-supervised learning for structured variables. In *NIPS*.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold regularization: a geometric framework for learning from examples. Technical Report TR-2004-06, University of Chicago Computer Science.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

Breese, J.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI'98*.

Chu, W., and Ghahramani, Z. 2005. Preference learning with gaussian processes. In *ICML'05*.

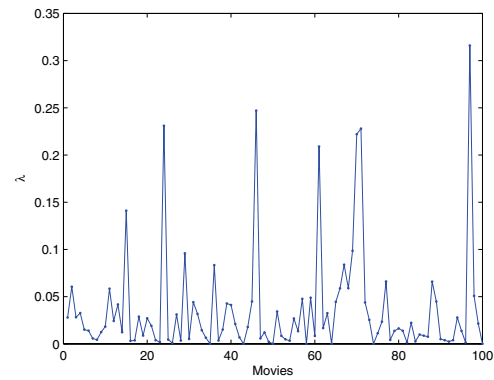Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. In *Annual of Eugenics*.

Figure 3: The λs computed for the 100 movies with 20 training users and 10 given ratings.

Freund, Y.; Lyer, R. D.; Schapire, R. E.; and Singer, Y. 1998. An efficient boosting algorithm for combining preferences. In *ICML'98*.

Goldberg, A.; Zhu, X.; and Wright, S. 2007. Dissimilarity in graph-based semi-supervised classification. In *AISTATS*.

Herbster, M.; Pontil, M.; and Wainer, L. 2005. Online learning over graphs. In *ICML '05*.

Hofmann, T. 2003. Gaussian latent semantic models for collaborative filtering. In *SIGIR'03*.

Joachims, T. 2003. Transductive learning via spectral graph partitioning. In *ICML'03*.

Kulis, B.; Basu, S.; Dhillon, I.; and Mooney, R. J. 2005. Semi-supervised graph clustering: A kernel approach. In *ICML'05*.

Pennock, D. M.; Horvitz, E.; Lawrence, S.; and Giles, C. L. 2000. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proc. UAI*.

Rennie, J., and Srebro, N. 2005. Fast maximun margin matrix factorization for collaborative prediction. In *ICML'05*.

Resnick, P.; Iacovou, N.; M.Sushak; Bergstrom, P.; and J.Riedl. 1994. Grouplense: An open architecture for collaborative filtering of netnews. In *Proc. Computer Supported Collaborative Work Conference*.

Srebro, N.; Rennie, J.; and Jaakkola, T. 2005. Maximun-margin matrix factorization. In *NIPS*.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. In *NIPS*.

Zhou, D.; Schölkopf, B.; and Hofmann, T. 2005. Semisupervised learning on directed graphs. In *NIPS*.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML'03*.