Personalized Resource Recommendations using Learning from Positive and Unlabeled Examples

Priyank Thakkar and K Kotecha

***Abstract*—This paper proposes a novel approach for recommending social resources using learning from positive and unlabeled examples. Bookmarks submitted on social bookmarking system delicious (http://www.delicious.com/) and artists on online music system last.fm (http://www.last.fm/) are considered as social resources. The foremost feature of this problem is that there are no labeled negative resources/examples available for learning a recommender/classifier. The memory based collaborative filtering has served as the most widely used algorithm for social resource recommendation. However, its predictions are based on some ad hoc heuristic rules and its success depends on the availability of a critical mass of users. This paper proposes model based two-step techniques to learn a classifier using positive and unlabeled examples to address personalized resource recommendation. In the first step of these techniques, naive Bayes classifier is employed to identify reliable negative resources. In the second step, to generate effective resource recommender, Classification and Regression Tree (CART) and Least Square-Support Vector Machine (LS-SVM) are exercised. A direct method based on LS-SVM is also put forward to realize the recommendation task. LS-SVM is customized for learning from positive and unlabeled data. Furthermore, the impact of feature selection on our proposed techniques is also studied. Memory based collaborative filtering as well as our proposed techniques exploit usage data to generate personalized recommendations. All the techniques are used for Top 5, Top 10 and Top 15 recommendations. Experimental results show that the proposed techniques outperform existing method appreciably. Among all the methods, customized LS-SVM performs the best.**

***Index Terms*—Learning from positive and unlabeled data, memory based collaborative filtering, personalized resource recommender.**

I. Introduction

SOCIAL bookmarking system is a web based resource sharing system that allows users to upload, share and organize their resources i.e. bookmarks and publications. The system has change the organization of bookmarks from an individual activity limited to a desktop to collective attempt over the web. Users can submit their resources that lead to large communities of users to collaboratively create accessible repositories of web resources. User bookmarking a resource (URL) on this system implicitly indicates his likings to the resource (URL). These resources are considered as positive examples of the user preference. Other resources (URLs) however do not imply a negative preference of the user on them. This leads to the situation where we have positive examples but no negative example for user preference.

Online music system such as last.fm constructs detailed profile of its user by analyzing details of the track the user listens to, either from internet radio stations, or the user's computer or many portable music devices. User listening to a particular artist many times is an implicit indicator about his positive preference about the artist. Artists who have not been listened by user however are not the indicators of user's negative preference. This again leads to the situation where we have positive examples but no negative example for user preference.

Conventional classification techniques require both labeled positive and labeled negative examples to build a recommender; they are thus not suitable to the problem.

The memory based collaborative filtering has served as the most widely used technique for resource recommendations but it has its own limitations of reliance on ad hoc heuristic rules and dependence of success on availability of a critical mass of users.

This paper proposes novel techniques to solve the problem. The first set of techniques employ naive Bayes method in the first step while Classification and Regression Tree (CART) and LS-SVM are utilized in the second step. The proposal is to first use naive Bayes classifier to extract some reliable negative resources from the unlabeled set and then apply CART or LS-SVM along with feature selection to build a recommender. The paper also put forwards a direct method based on LS-SVM to build a resource recommender. LS-SVM has been tailored to learn from positive and unlabeled data. Experimental results show that the proposed techniques outperform existing method significantly.

II. Related Work

The task of resource recommendation entails retrieving and recommending interesting items - in our study, bookmarks or artists - to the user. Recommendations can be based on a range of information sources about the user and the resource, contributing information at different representation levels. One such level of representation, usage data, is the set of transaction patterns that shows which resources have been bookmarked/listened, and which users have bookmarked/listened to them. In these systems, tags present an additional level of representation, linking users to resource through an alternative route. The past few years have seen an escalating number of approaches for resource recommendation that exploits these two types of data representations [1]–[5].

P. Thakkar is with the Department of Computer Science and Engineering, Institute of Technology, Nirma University, Gujarat, India. e-mail: (priyank.thakkar@nirmauni.ac.in).

K. Kotecha was with the Institute of Technology, Nirma University, Gujarat, India. He is now with Parul University, Waghodia, Gujarat, India e-mail: (drketankotecha@gmail.com).

Usage, tag and metadata of resource were used in [6] and [5] for recommendation generation. They also investigated about how to fuse together different recommendation approaches to further improve recommendation accuracy. These approaches typically used a memory based Collaborative Filtering (CF) algorithm to make their recommendations. There are distinct research efforts [7], [8] whose focus is to recommend tags for social resources but the focus of this paper is limited to to recommending social resources.

A theoretical study of PAC learning from positive and unlabeled examples under the statistical query model [9] was presented in [10]. In [11], Liu et al. proposed a method called S-EM to classify text documents using positive and unlabeled examples. This method combined naive Bayesian classifier and EM algorithm [12]. Some new methods for identifying set of reliable negative documents and classifying text documents were proposed by Liu et al. in [13]. Xiaoli Li et al. in [14], proposed a method that combined Rocchio's technique with SVM to build a text classifier. Learning from positive and unlabeled examples (PU learning) was also used in [15] and [16] for named entity disambiguation in streaming data and learning gene regularity networks respectively.

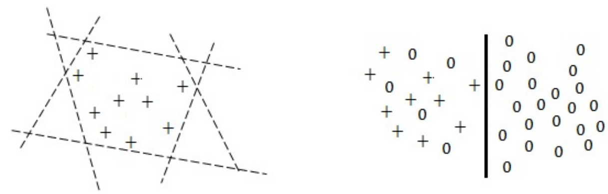
To the best of our effort, a review of the current literature unveils that PU learning has not been employed for social resource recommendation till date. In this paper, social resources are recommended to the user by employing PU learning in conjunction with feature selection.

III. Collaborative Filtering for Resource Recommendation

Algorithms which use usage data for recommendation purposes are referred to as collaborative filtering algorithms. In [6], Toine Bogers et al. extended one specific CF algorithm: the k-Nearest Neighbor (k-NN) algorithm. In their proposal, first step involved locating users that were most similar to the active user, i.e., the user to whom the new items were to be recommended. Similarity between the active user and all other users in the system was calculated by considering the overlap in resources they had preferred. Each user u_k was represented as a unary user profile vector u_k that represented all the resources that are preferred by him/her with a 1. Cosine similarity metric was used to determine the similarity between two users u_k and the active user u_a as $sim_{cosine}(u_a, u_k) = \frac{u_a \cdot u_k}{||u_a||||u_k||}$. Cosine similarity had been used effectively with data sets with implicit ratings [17].

In the second step, the resources of the most like-minded users were gathered to determine the most suitable recommendations for the active user. The supposition here was, more similar two users were in the resources they share, the more of similar mind they were. The top k most similar users for the active user u_a formed the Set of Similar Users $SSU(u_a)$. Taking into consideration this set of the nearest neighbors, the final prediction score $S_{a,l}$ for each of the SSU's resources i_l was computed as $S_{a,l} = \sum_{u_k \in SSU(u_a)} sim_{cosine}(u_a, u_k)$.

Thus, the predicted score of a resource i_l was the sum of similarity values (between 0 and 1) of all N nearest neighbors that actually posted resource i_l.



(a) With only positive examples (b) with both positive and unlabeled examples

Fig. 1: Usefulness of Unlabeled Data [18]

Lastly, all resources were ranked by their predicted score $S_{a,l}$. Resources already posted by the active user were filtered out to generate the final list of recommendations for the active user.

IV. Learning from Positive and Unlabeled Examples

In this section, the usefulness of unlabeled data and theoretical foundations of learning from positive and unlabeled examples is discussed as convoluted in [11] and [18].

A. *Usefulness of Unlabeled Data*

In this segment, some perception on why learning from positive and unlabeled examples is feasible and why unlabeled examples are useful is built up [18]. Figure 1 depicts the idea.

In Figure 1a, only positive resources/examples are represented. Symbol '$+$' is used to represent the positive resources. It is assumed that a linear classifier is adequate for the classification. In this scenario, it is difficult to identify where to draw the line that separates positive and negative examples because it is not known where the negative examples might be. There are infinite possibilities. Conversely, if the unlabeled data are added to the space as shown in Figure 1b, it becomes very apparent where the separation line should be. Symbol '0' is used to represent unlabeled data.

B. *Theoretical Foundations of PU Learning*

Let (x_i, y_i) be random variables drawn independently from probability distribution $I_{(x,y)}$ where $y \in \{-1, 1\}$ is the conditional random variable that is to be approximated given x. x_i is used to represent a resource while y_i is used to represent its class. Positive resources can be symbolized by class '$+1$' while '-1' can be used to characterize negative resources. Assume that the positive and unlabeled resources are independently drawn from the conditional distribution $I_{x|y}$ and the marginal distribution I_x. Aim is to learn a classification function f, that can separate positive and negative resources with minimum probability of error, $Pr(f(x) \neq y)$. Rewriting it into more useful form,

$$Pr(f(x) \neq y) = Pr(f(x) = 1 \;\; and \;\; y = -1) + Pr(f(x) = -1 \;\; and \;\; y = 1) \qquad (1)$$

The first expression in Equation 1 can be further expressed as

$$
\begin{aligned}
Pr(f(x) = 1 \ \ and \ \ y = -1) &= Pr(f(x) = 1) \\
&\quad - Pr(f(x) = 1 \ \ and \ \ y = 1) \\
= Pr(f(x) = 1) - (Pr(y = 1) &- Pr(f(x) = -1 \\
&\quad and \ \ y = 1))
\end{aligned} \tag{2}
$$

Substituting results from Equation 2 into Equation 1, we get,

$$
\begin{aligned}
Pr(f(x) \neq y) &= Pr(f(x) = 1) - Pr(y = 1) \\
&\quad + 2Pr(f(x) = -1|y = 1)Pr(y = 1)
\end{aligned} \tag{3}
$$

Since $Pr(y = 1)$ is constant, the probability of error can be minimized by minimizing

$$Pr(f(x) = 1) + 2Pr(f(x) = -1|y = 1)Pr(y = 1) \tag{4}$$

If $Pr(f(x) = -1|y = 1)$ can be held small, minimizing the probability of error is same as minimizing $Pr(f(x) = 1)$. This is just about same as minimizing $Pr_U(f(x) = 1)$ (because number of positive examples are very small compared to number of unlabeled examples) while holding $Pr_P(f(x) = 1) \geq r$ where r is the recall, i.e. $Pr(f(x) = 1|y = 1)$ and U and P are set of unlabeled and positive examples respectively. Note that $(Pr_P(f(x) = 1) \geq r)$ is same as $(Pr_P(f(x) = -1) \leq (1 - r))$.

Above formulations allow to model the problem as constrained optimization problem where, the objective is to minimize the number of unlabeled examples labeled as positive, subject to the constraint that the fraction of errors on the positive examples is no more than $1 - r$ [18].

V. THE PROPOSED TWO STEP TECHNIQUES

In the previous section, it has been shown theoretically that by using positive and unlabeled resource sets, accurate classifiers can be built with high probability provided that sufficient positive and unlabeled resources are available. Nevertheless, the discussed theoretical method has two drawbacks: (1) The constrained optimization problem may not be easy to solve (2) Given a practical problem, it seems to be difficult to select a preferred recall level that will give a good classifier. This section proposes a practical two step heuristic techniques inspired from the work in [13], [18]. In the first step of the techniques, reliable negative resources are extracted from the unlabeled set using nave Bayesian method. In the second step, CART and LS-SVM are experimented in conjunction with feature selection (by means of Information Gain method [19]) to build the recommender. Lastly the built recommender is used to generate personalized recommendations.

1) Finding Reliable Negative Resources: Naive Bayesian method is one of the popular techniques for classification. Even though the postulation that features are independent given class label of a resource is not realistic in this domain, it has been shown to perform very well in practice by many researchers [20], [21].

Given a set of training resources I, each resource i is considered as a vector of n attribute values $[i.e., i = (i_1, i_2, i_3, \cdots, i_n)]$. In this work, each attribute represents a distinct user and the value of the attribute indicates whether the corresponding user has liked or disliked the corresponding resource. This means, all the attributes are modelled as Boolean attributes. This allows us to fit the multivariate Bernoulli distribution to the data. The naive Bayesian method decides the class C of resource i as the one that maximizes the conditional probability $P(C|i)$. According to Bayes' rule,

$$P(C|i) = \frac{P(i|C)P(C)}{P(i)} \tag{5}$$

To determine $P(i|C)$, naive Bayesian assumption, that attributes are statistically independent is made. As i is a vector of n attribute values, this supposition leads to,

$$P(i|C) = P(i_1, i_2, i_3, \cdots, i_n|C) = \prod_{k=1}^{n} P(i_k|C) \tag{6}$$

The proportion of resources from class C that includes attribute value i_k is used to calculate each $P(i_k|C)$. $P(C)$ is the probability of resources for class C, which is calculated as the fraction of the training resources that fall in class C. $P(i)$ is common denominator, which is not required in the calculation as only the class label is to be decided. Laplacian prior is also used in the actual calculation of conditional probability to avoid probability estimation to 0.

To identify set of reliable negative (RN) resources from the unlabeled set U, steps shown in Algorithm 1 are followed.

Algorithm 1 Algorithm to identify reliable negative (RN) resources

Input: positive set P, unlabeled set U
Output: reliable negative set RN

$RN \leftarrow \emptyset$
Assign the class label 1 to each resource in positive set P
Assign the class label -1 to each resource in unlabeled set U
Build a naive Bayes classifier using P and U. Fit multivariate Bernoulli distribution to the data
for all resource $r \in U$ **do**
  if its probability $P(1|r) < 0.5$ **then**
    $RN = RN \cup \{r\}$
  end if
end for
return RN

A. Building and using Recommender to make Predictions

Resources which are part of P and RN form the data for building the recommender. Before this data is used to build the recommender, it is partitioned into training and test set. Information Gain [19] is also used to identify those features which are important. Data with only important features are used to learn the recommender. The number of important features is varied in the experiment to study its impact on the final outcome. CART [19] and LS-SVM are used to build recommender in second step. Performance of these methods is also compared.

Decision tree learners build a decision tree by recursively partitioning examples into subgroups until those subgroups include examples of a single class. A partition is formed by a test on the selected attribute. In this study, CART is used to construct a regression tree. CART uses Gini index which selects the attribute that maximizes the reduction in impurity [19].

In SVM, the idea is to identify maximum marginal hyper plane $< w \cdot x > + b = 0$ to separate the data belonging to different class y_i where $y_i \in \{1, -1\}$. Here, a direction perpendicular to the hyper plane is defined by w, $b \in R$ is a bias and x is a set of input vectors. $w \cdot x$ is a dot product of w and x. In classical SVM, one needs to solve convex quadratic programming problem. LS-SVM solves set of linear equations instead of a convex quadratic programming problem for classical SVMs [22]. Linear LS-SVM under separable case is the best condition. In practice, though, the training data is more or less always noisy. It has been shown by means of extensive empirical studies that LS-SVM is comparable to SVM in terms of generalization performance [23], [24]. However, the underlying reason for their similarity is not well understood yet [25]. In this study, LS-SVM is considered under non-separable case.

Definition (Linear LS-SVM: Non-Separable Case): Given a set of training examples which are linearly separable, $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, learning is to solve the following constrained minimization problem,

$$\begin{aligned} Minimize: &\frac{< w \cdot w >}{2} + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \\ Subject\ \ to: &\, y_i(< w \cdot x_i > +\ b) = 1 - \xi_i, \\ &i = 1, 2, \ldots, n \end{aligned} \tag{7}$$

where $C \geq 0$ is user defined parameters and ξ_i is a slack variable [18].

Solving the constrained minimization problem in Equation 7 produces the solutions for w and b, which in turn give us the maximum margin hyper plane $< w \cdot x_i > +\ b = 0$ with the margin $\frac{2}{||w||}$.

In a nutshell, the procedure to be followed, for the second step of two step techniques, is depicted in Algorithm 2.

Algorithm 2 Algorithm for building the recommender

Input: positive set P, reliable negative set RN, number of features N to be considered to build the recommender, choice CH to specify the method to be used in second step to build the recommender

Output: Built Recommender R

1: Assign the class label 1 to each resource in positive set P

2: Assign the class label -1 to each resource in reliable negative set RN

3: Construct training and testing set.

4: Identify N most important features using Information Gain on the training data

5: Build recommender R (using the training set and selected N features) through the method specified by means of the value of CH

6: Return Recommender R

Once the personalized recommender is built for the user, it can be used for that user to predict the confidence by which each of the existing resource belongs to positive set.

VI. THE PROPOSED DIRECT METHOD

In this section, a direct method is proposed, inspired from the work in [13], [18]. Direct method is based on LS-SVM and eliminates the need of identifying reliable negative resources. Let the set of training examples be $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, where x_i is the input vector and y_i is its corresponding class label and $y_i \in \{1, -1\}$. Presume that first $k - 1$ examples are positive examples $(+1)$, while the rest are unlabeled examples, which are labelled as negative (-1). Thus, the negative set has noise, i.e., contains positive examples. In practice, the positive set may also contain some noise. If noise is allowed in positive examples, then learning is to solve the following constrained optimization problem.

$$\begin{aligned} Minimize: &\frac{< w \cdot w >}{2} + \frac{C_+}{2}\sum_{i=1}^{k-1} \xi_i^2 + \frac{C_-}{2}\sum_{i=k}^{n} \xi_i^2 \\ Subject\ \ to: &\, y_i(< w \cdot x_i > + b) = 1 - \xi_i, \quad i = 1, 2, \ldots, n \end{aligned} \tag{8}$$

where $C_+, C_- \geq 0$ are user defined parameters and ξ_i is a slack variable. C_+, C_- can be varied to achieve the objective. Intuitively, a bigger value is assigned to C_+ while a smaller value to C_- because the unlabeled set, which is assumed to be negative, may contain positive data. In all the experiments performed in this study, C_+ and C_- are set to 0.9 and 0.1 respectively on the basis of empirical evidence. Lagrangian Primal corresponding to the optimization problem with equality constraints in Equation 8 is

$$\begin{aligned} L_p = &\frac{1}{2} < w \cdot w > + \frac{C_+}{2}\sum_{i=1}^{k-1} \xi_i^2 + \frac{C_-}{2}\sum_{i=k}^{n} \xi_i^2 \\ &+ \sum_{i=1}^{n} \alpha_i [1 - \xi_i - y_i(< w \cdot x_i > + b)], \quad \alpha_i \in R \end{aligned} \tag{9}$$

As discussed in [26], conditions for the optimal solution give us following expressions:

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (10)$$

$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \quad (11)$$

$$\frac{\partial L_p}{\partial \xi_i} = 0 \Rightarrow C_+\xi_i - \alpha_i = 0 \Rightarrow \xi_i = \frac{\alpha_i}{C_+}, \quad i = 1, 2, \ldots, k-1 \quad (12)$$

$$\frac{\partial L_p}{\partial \xi_i} = 0 \Rightarrow C_-\xi_i - \alpha_i = 0 \Rightarrow \xi_i = \frac{\alpha_i}{C_-}, \quad i = k, k+1, \ldots, n \quad (13)$$

$$\frac{\partial L_p}{\partial \alpha_i} = 0 \Rightarrow y_i(< w \cdot x_i > +b) - 1 + \xi_i = 0, \quad i = 1, 2, \ldots, n \quad (14)$$

Eliminating w and ξ_i from Equation 14 using Equations 10 and 12 and then using this result along with Equation 11 gives linear system as shown in Equation 15 ,

$$\begin{bmatrix} 0 & Y_{k-1}^T \\ Y_{k-1} & \Omega_{k-1} + C_+^{-1} I_{k-1} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_{k-1} \end{bmatrix} \quad (15)$$

where, $Y_{k-1} = [y_1, y_2, \ldots, y_{k-1}], 1_{k-1} = [1, 1, \ldots, 1], \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_{k-1}]$. I_{k-1} is $(k-1) \times (k-1)$ identity matrix and $\Omega \in R^{(k-1)\times(k-1)}$ is the kernel matrix defined by $\Omega_{ij} = y_i y_j < x_i \cdot x_j >$. $i, j = 1, 2, \ldots, k-1$. Eliminating w and ξ_i from Equation 14 using Equations 10 and 13 and then using this result along with Equation 11 gives the linear system as shown in Equation 16,

$$\begin{bmatrix} 0 & Y_{n-(k-1)}^T \\ Y_{n-(k-1)} & \Omega_{n-(k-1)} + C_-^{-1} I_{n-(k-1)} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_{n-(k-1)} \end{bmatrix} \quad (16)$$

where, $Y_{n-(k-1)} = [y_k, y_{k+1}, \ldots, y_n], 1_{n-(k-1)} = [1, 1, \ldots, 1], \alpha = [\alpha_k, \alpha_{k+1}, \ldots, \alpha_n]$. $I_{n-(k-1)}$ is $(n-(k-1)) \times (n-(k-1))$ identity matrix and $\Omega \in R^{(n-(k-1))\times(n-(k-1))}$ is the kernel matrix defined by $\Omega_{ij} = y_i y_j < x_i \cdot x_j >$. $i, j = k, k+1, \ldots, n$.

Solving the linear systems in 15 and 16 give values of α_i for $i = 1, 2, \ldots, k-1$ and $i = k, k+1, \ldots, n$ respectively in addition to the value of b. This in turn, gives the maximum margin hyper plane $< w \cdot x_i > +b = 0$ with the margin $\frac{2}{||w||}$.

VII. EMPIRICAL EVALUATION

In this section our proposed techniques are evaluated and compared with memory based collaborative filtering which is the most widely used technique for social resource recommendations.

A. Dataset

Two data sets are used in all the experiments. The first one is the Hetrec2011-delicious-2k (http://www.delicious.com/) and the second one is Hetrec2011-lastfm-2k (http://www.last.fm/). Both the data sets were released in the framework of the 2^{nd} International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011) (http://ir.ii.uam.es/hetrec2011/) at the 5th ACM Conference on Recommender Systems (RecSys 2011) [27] (http://recsys.acm.org/2011/).

In $delicious$ data set, 7 files are given in addition to the readme file. Usage data is extracted from user_taggedbookmarks.dat file. This extraction resulted in to 1867 unique users and 69226 unique URLs. Each user is modelled as Boolean feature vector where the number of features is 69226. A value 1 of a specific feature in feature vector of the user indicates that the user has bookmarked the corresponding URL. In $lastfm$ data set, 6 files are given in addition to the readme file. Listen count of users is extracted from user_artists.dat. There are 1892 unique users and 17632 unique artists. If the user had listened to a specific artist more than some listen count threshold (one, in this study) times, it is considered that the user has positive preference for that artist, else he has negative preference. This consideration resulted in Boolean user feature vectors.

B. Experimental Methodology

In the experiments, techniques proposed in this study and a typical memory based collaborative filtering as described in [6] are evaluated. Twenty test users who have bookmarked at least 45 URLs from the $delicious$ data set are selected for experiments on $delicious$ data set. Similarly twenty test users who have shown positive preference (based on listen count threshold) for at least 45 artists from the $lastfm$ data set are selected for the experiments on $lastfm$ data set. For each data set, stratified 3-fold cross validation is carried out. In both the approaches after calculating predicted score (using the built recommender) for each of the resources in the test set, they are arranged in descending order and recommended from the top of the list based on Top 5, Top 10 and Top 15 recommendations. Experiments are also carried out by changing neighborhood size in memory based collaborative filtering and selecting different number of features in the proposed techniques to study the impact of these parameters on the quality of recommendations.

C. Evaluation Measures

Let I_T be an evaluation data set consisting of $|I_T|$ examples (x_i, Y_i), $i = 1, 2, \cdots, |I_T|$, $Y_i \in \{-1, 1\}$. Let f be the recommender and $Z_i = f(x_i)$ be the prediction by f for example x_i. In these experiments, Precision, Recall and F-measure for the recommender f on the test data set I_T are calculated as follows.

$$Precision(f, I_T) = \frac{1}{|I_T|} \sum_{i=1}^{|I_T|} \frac{|Y_i \cap Z_i|}{|Z_I|} \quad (17)$$

$$Recall(f, I_T) = \frac{1}{|I_T|} \sum_{i=1}^{|I_T|} \frac{|Y_i \cap Z_i|}{|Y_I|} \quad (18)$$

$$F - measure(f, I_T) = \frac{1}{|I_T|} \sum_{i=1}^{|I_T|} \frac{2|Y_i \cap Z_i|}{|Z_I| + |Y_I|} \quad (19)$$

Precision(P) @ TopN, Recall(R) @ TopN and F-measure(F) @ TopN are used as the performance measures. They are calculated by considering only the topmost results returned by the classifier in the Equations 17, 18 and 19.

D. Results and Discussions

This section discusses about the experimental results. Tables I and II show the results on $delicious$ and $lastfm$ data sets respectively. Results for memory based collaborative filtering, proposed two step techniques and proposed direct technique are shown.

Experiments are carried out with varying number of nearest users in case of memory based collaborative filtering and varying number of features in case of the proposed techniques. However, tables show results for only that value of number of Nearest Neighbors (NN)/features (NOF), where individual technique has performed the best overall.

It is apparent from the experimental results that the proposed techniques produce significantly better results than memory based collaborative filtering which is the most widely used technique for resource recommendations. It is also evident that the best results are achieved when number of Nearest Neighbor (NN) is set to 30 and 10 for $delicious$ and $lastfm$ data sets respectively, in case of collaborative filtering. The proposed techniques perform best at 1000 features for $delicious$ data set while, they perform best at 10 (CART) and 15 (LS-SVM and direct approach) features in case of $lastfm$ data set.

Figures 2 and 3 illustrate the impact of number of nearest neighbors used to make predictions in memory based collaborative filtering approach.

Figures 4, 5 and 6 depict significance of Nnumber of Features (NOF) used to generate TOP 5, Top 10 and TOP 15 recommendations respectively through these proposed techniques. These figures show results for $delicious$ data set.

Similarly, results for $lastfm$ data set are shown in Figures 7, 8 and 9. It is apparent from the figures that memory based techniques as well as the proposed techniques are sensitive to the number of nearest neighbors/features.

It is palpable that selecting the right value for this parameter definitely affects the performance of the recommender.

While learning from positive and unlabeled examples, it is assumed that unlabeled examples contain errors. In practice, the positive set (set of positive examples) may also contain some errors. This should be considered while learning a classifier from positive and unlabeled examples. In two step techniques, there is no way to incorporate this consideration and all positive examples are considered noise free. However, as stated earlier, positive examples may also have errors and these erroneous examples may degrade the quality of learnt model through two step techniques which in turn can degrade the performance of two step techniques.

In regular LS-SVM, regularization parameter C is used to weigh errors. There is no way to weigh positive and negative errors differently. In modified LS-SVM, proposed in this paper, two regularization parameters C_+ and C_- are introduced to weigh positive and negative errors differently. These parameters can be controlled individually and thereby positive and negative errors can be controlled individually while learning the LS-SVM model. This is the reason behind the improved results of modified LS-SVM.

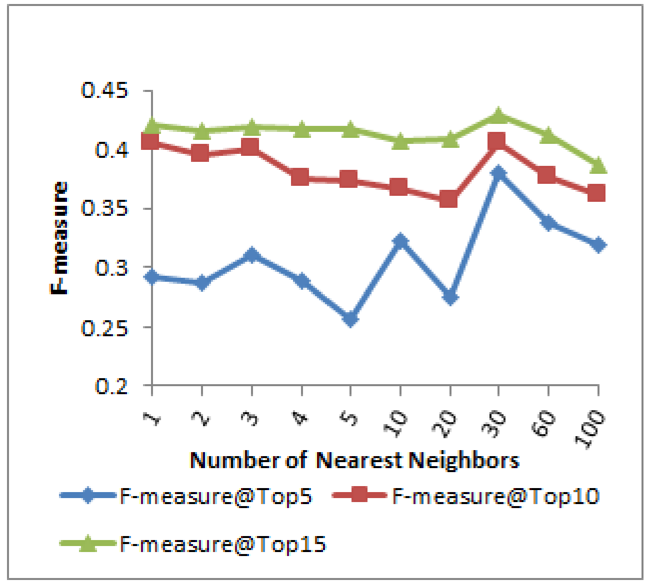


Fig. 2: Collaborative filtering - $delicious$ data set

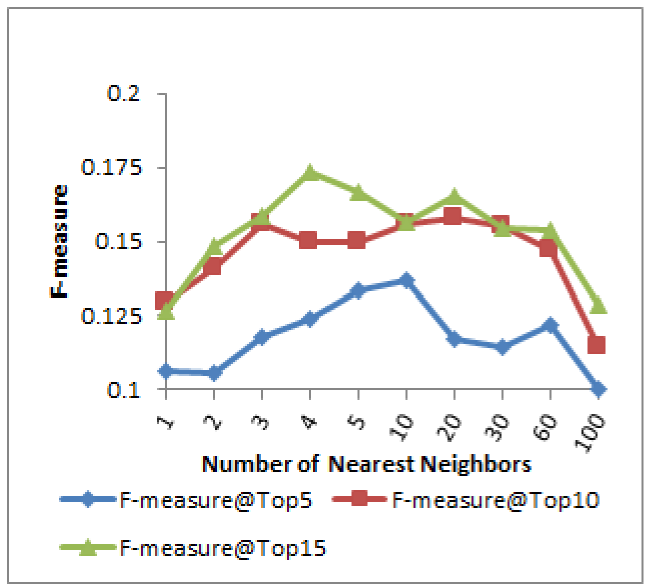


Fig. 3: Collaborative filtering - $lastfm$ data set

TABLE I: Results on *delicious*

P/R/F @TopN	**Collaborative Filtering (NN=30)**	**Proposed Two Step Technique (CART in Second Step) (NOF=1000)**	**Proposed Two Step Technique (LS-SVM in Second Step) (NOF=1000)**	**Proposed Direct Method Based on Modified LS-SVM (NOF=1000)**
P/R/F @Top5	0.5267/0.2980/ 0.3806	0.7400/0.4116/ 0.5290	0.7933/0.4522/ 0.5760	0.8200/0.4697/ 0.5973
P/R/F @Top10	0.3900/0.4202/ 0.4045	0.5867/0.6263/ 0.6059	0.6000/0.6495/ 0.6238	0.6933/0.7594/ 0.7249
P/R/F @Top15	0.3489/0.5596/ 0.4298	0.4578/0.7018/ 0.5541	0.4511/0.7090/ 0.5514	0.5311/0.8377/ 0.6501

TABLE II: Results on *lastfm*

P/R/F @TopN	**Collaborative Filtering (NN=10)**	**Proposed Two Step Technique (CART in Second Step) (NOF=10)**	**Proposed Two Step Technique (LS-SVM in Second Step) (NOF=15)**	**Proposed Direct Method Based on Modified LS-SVM (NOF=15)**
P/R/F @Top5	0.2778/0.0906/ 0.1366	0.3067/0.1010/ 0.1520	0.3800/0.1231/ 0.1860	0.4067/0.1320/ 0.1993
P/R/F @Top10	0.1978/0.1289/ 0.1561	0.2400/0.1575/ 0.1902	0.2600/0.1680/ 0.2041	0.3100/0.2022/ 0.2448
P/R/F @Top15	0.1578/0.1547/ 0.1562	0.1867/0.1832/ 0.1849	0.2022/0.1961/ 0.1991	0.2467/0.2426/ 0.2446

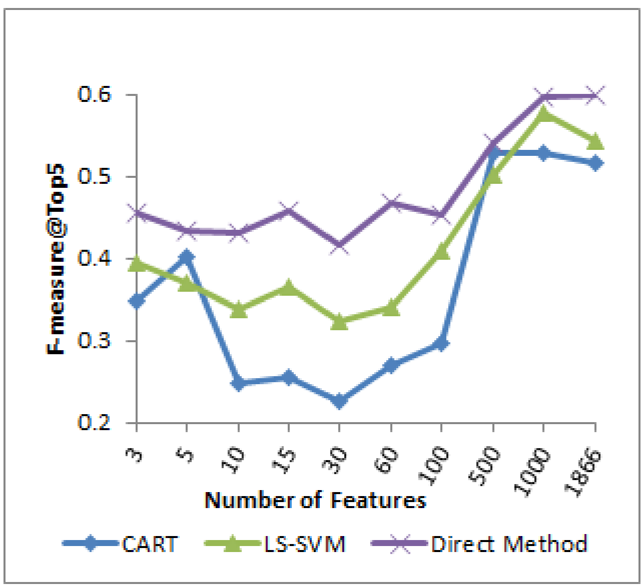


Fig. 4: F-measure @ Top5 - *delicious* data set

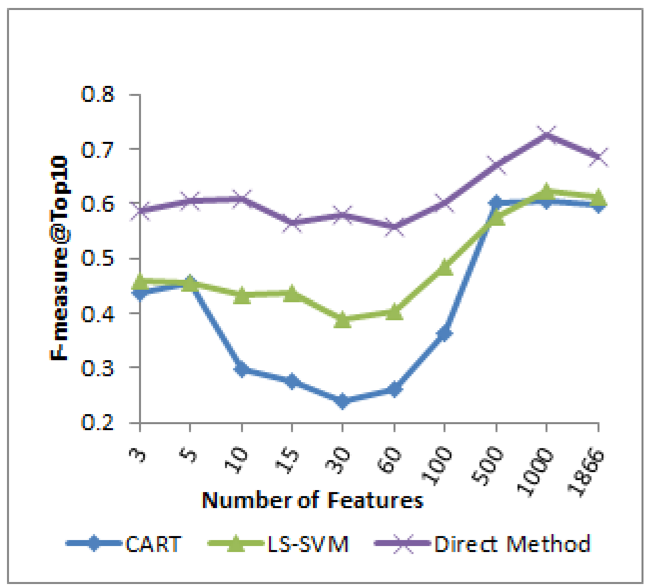


Fig. 5: F-measure @ Top10 - *delicious* data set

VIII. Conclusions

In this paper, the problem of personalized resource recommendation under the situation where only positive and unlabeled examples are available is discussed. Methods based on naive Bayes classifier and CART/LS-SVM to learn a recommender using positive and unlabeled (PU) examples are proposed.

Moreover, a direct method in which LS-SVM is adapted to learn from PU examples is also proposed. The proposed methods also use feature selection to its advantage. Experimental results show that all the proposed techniques perform considerably better than memory based collaborative filtering. Direct method performs the best among all the techniques discussed.

It is furthermore inferred that selecting right number of features definitely affects the accuracy of the recommender.

The tailoring of LS-SVM to enable it learn from positive and unlabeled examples is a unique contribution of this work. To the best of our knowledge, this is the first attempt which models the social resource recommendation as learning from the positive and unlabeled examples.

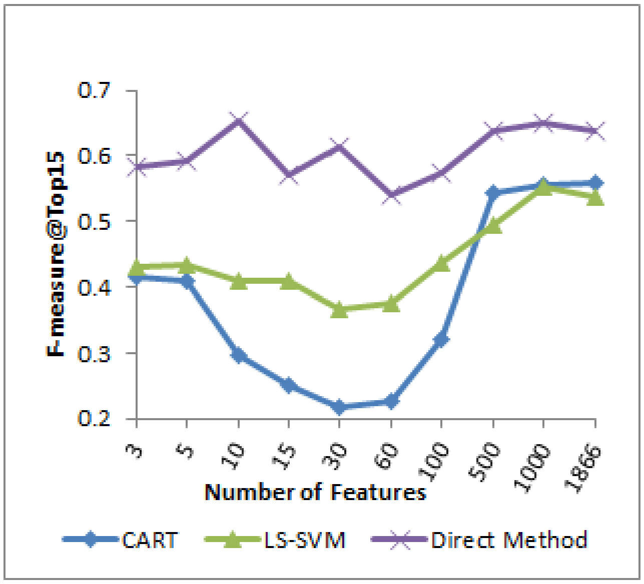


Fig. 6: F-measure @ Top15 - $delicious$ data set

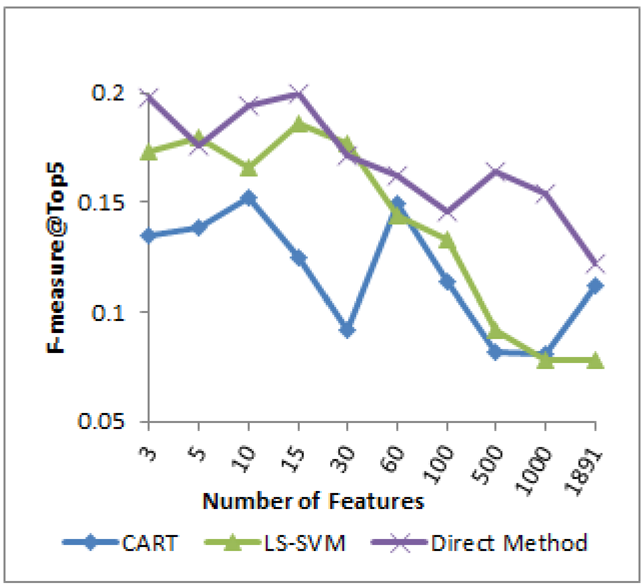


Fig. 7: Fmeasure @ Top5 - $lastfm$ data set

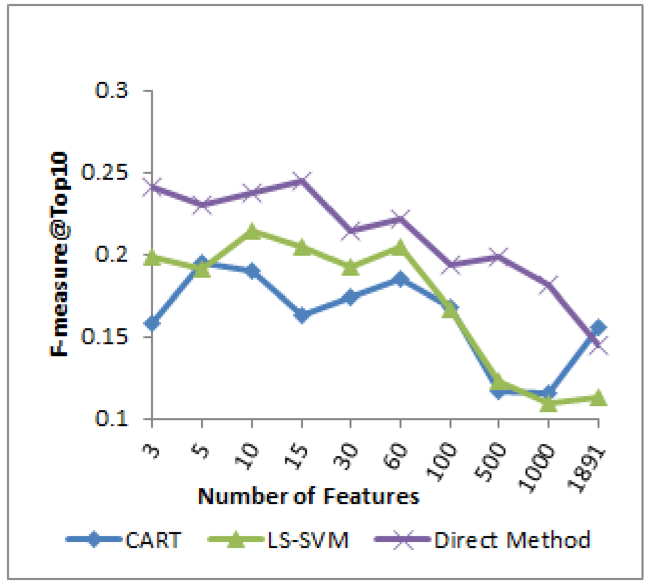


Fig. 8: Fmeasure @ Top10 - $lastfm$ data set

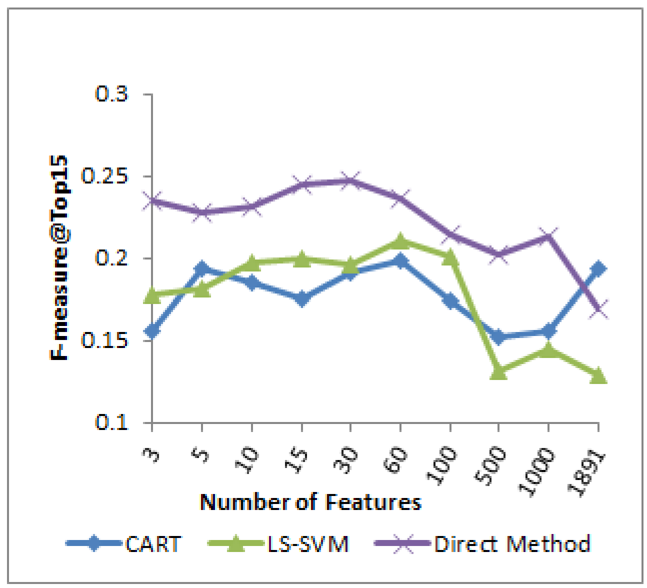


Fig. 9: Fmeasure @ Top15 - $lastfm$ data set

REFERENCES

[1] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "Information retrieval in folksonomies: Search and ranking," in *The semantic web: research and applications*. Springer, 2006, pp. 411–426.

[2] M. Clements, A. P. de Vries, and M. J. Reinders, "Optimizing single term queries using a personalized markov random walk over the social graph," in *Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR)*, 2008.

[3] K. H. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme, "Tag-aware recommender systems by fusion of collaborative filtering algorithms," in *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008, pp. 1995–1999.

[4] R. Wetzker, W. Umbrath, and A. Said, "A hybrid approach to item recommendation in folksonomies," in *Proceedings of the WSDM'09 Workshop on Exploiting Semantic Annotations in Information Retrieval*. ACM, 2009, pp. 25–29.

[5] R. Patel, P. Thakkar, and K. Kotecha, "Enhancing movie recommender system," *International Journal of Advnaced Research in Engineering and Technology*, vol. 5, no. 1, pp. 73–82, 2014.

[6] T. Bogers and A. Van Den Bosch, "Fusing recommendations for social bookmarking web sites," *International Journal of Electronic Commerce*, vol. 15, no. 3, pp. 31–72, 2011.

[7] A. Marchetti, M. Tesconi, F. Ronzano, M. Rosella, and S. Minutoli, "Semkey: A semantic collaborative tagging system," in *Workshop on Tagging and Metadata for Social Information Organization at WWW*, vol. 7, 2007, pp. 8–12.

[8] S. Yagnik, P. Thakkar, and K. Kotecha, "Recommending tags for new resources in social bookmarking system," *International Journal of Data Mining & Knowledge Management Process*, vol. 4, no. 1, pp. 19–32, 2014.

[9] M. Kearns, "Efficient noise-tolerant learning from statistical queries," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 983–1006, 1998.

[10] F. Denis, "Pac learning from positive statistical queries," in *Algorithmic Learning Theory*. Springer, 1998, pp. 112–126.

[11] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *ICML*, vol. 2. Citeseer, 2002, pp. 387–394.

[12] A. P. Dempster, N. M. Laird, D. B. Rubin *et al.*, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.
[13] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabelled examples," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 179–186.
[14] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *IJCAI*, vol. 3, 2003, pp. 587–592.
[15] A. Davis, A. Veloso, A. S. da Silva, W. Meira Jr, and A. H. Laender, "Named entity disambiguation in streaming data," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 815–824.
[16] L. Cerulo, C. Elkan, and M. Ceccarelli, "Learning gene regulatory networks from only positive and unlabeled data," *Bmc Bioinformatics*, vol. 11, no. 1, p. 228, 2010.
[17] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.
[18] B. Liu, *Web data mining*. Springer, 2007.
[19] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. Morgan kaufmann, 2006.
[20] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2-3, pp. 103–130, 1997.
[21] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752. Citeseer, 1998, pp. 41–48.
[22] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.
[23] T. Van Gestel, J. A. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and J. Vandewalle, "Benchmarking least squares support vector machine classifiers," *Machine Learning*, vol. 54, no. 1, pp. 5–32, 2004.
[24] P. Zhang and J. Peng, "Svm vs regularized least squares classification," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1. IEEE, 2004, pp. 176–179.
[25] J. Ye and T. Xiong, "Svm versus least squares svm," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 644–651.
[26] S. S. Rao and S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.
[27] I. Cantador, P. Brusilovsky, and T. Kuflik, "2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011)," in *Proceedings of the 5th ACM conference on Recommender systems*, ser. RecSys 2011. New York, NY, USA: ACM, 2011.

Priyank Thakkar Dr. Priyank Thakkar is an associate professor at Computer Science and Engineering department, Institute of Technology, Nirma University, Gujarat, India. He has done B.E. from Sarvajanik College of Engineering & Technology, Surat in the year 2000. He has done M.E. from BVM Engineering College, V. V. Nagar in the year 2008. He has more than 15 years of teaching experience. His area of interest includes Data and Web Mining, Machine Learning, Soft Computing and Image Processing. He has published more than 10 research papers in international journals and conferences.

K Kotecha Dr. Ketan Kotecha is a passionate educationalist and believes in becoming the catalyst to bring the change we want to see. Founding Provost (Vice Chancellor) of the Parul University, Gujarat, India, he has Doctorate from IIT Bombay- one of the most prestigious of all IITs. Dr Kotecha has rare ability of achieving the balance between the academics, administration and research. His research interests include Artificial Intelligence, Computer Algorithms & Machine Learning, and Higher Order Thinking skills, Critical Thinking and Ethics and Value. He has more than 100 papers published/presented in international conferences around the world, to his credit and 3 patents filling. He has guided no. of students on diverse field of Computer Engineering and especially in Machine Learning. Previously, Dr Kotecha was Director - Academic Development and Research Cell, Nirma University, Ahmedabad and also Dean and Director, Institute of Technology, Nirma University.