Prediction with Confidence in Item Based Collaborative Filtering

Tadiparthi V.R. Himabindu[1], Vineet Padmanabhan[1](✉), Arun K. Pujari[1], and Abdul Sattar[2]

[1] School of Computer and Information Sciences, University of Hyderabad, Hyderabad, India
himaworld_06@yahoo.com, {vineetcs,akpcs}@uohyd.ernet.in

[2] Griffith University, Brisbane, Australia
a.sattar@griffith.edu.au

Abstract. Recommender systems can be viewed as prediction systems where we can predict the ratings which represent users' interest in the corresponding item. Typically, items having the highest predicted ratings will be recommended to the users. But users do not know how *certain* these predictions are. Therefore, it is important to associate a confidence measure to the predictions which tells users how certain the system is in making the predictions. Many different approaches have been proposed to estimate confidence of predictions made by recommender systems. But none of them provide guarantee on the error rate of these predictions. Conformal Prediction is a framework that produces predictions with a guaranteed error rate. In this paper, we propose a conformal prediction algorithm with item-based collaborative filtering as the underlying algorithm which is a simple and widely used algorithm in commercial applications. We propose different nonconformity measures and empirically determine the best nonconformity measure. We empirically prove validity and efficiency of proposed algorithm. Experimental results demonstrate that the predictive performance of conformal prediction algorithm is very close to its underlying algorithm with little uncertainty along with the measures of confidence and credibility.

Keywords: Recommender systems · Conformal prediction · Confidence · Nonconformity measure

1 Introduction

Collaborative filtering (CF) is a very promising approach in recommender systems and is the most widely adopted technique both in academic research and commercial applications. CF algorithms can be classified in two ways: in *neighborhood based approaches* prediction and recommendation can be done either by computing the similarities between users (user-based collaborative filtering (UBCF) [16]) or similarities between items (item-based collaborative filtering

DOI: 10.1007/978-3-319-42911-3_11

(IBCF) [2]) and *model-based approaches* [17] use mathematical models for making predictions. Many model-based algorithms are very complex which involves estimation of large number of parameters. Moreover if the assumptions of the model do not hold, it may lead to wrong predictions. On the other hand, neighborhood approaches are very simple both in terms of underlying principles and implementation while achieving reasonably accurate results. But UBCF does not perform well when the active user is having too few neighbors and neighbors with very low correlation to the active user [8]. In our paper, for the proposed conformal prediction algorithm we have chosen IBCF as the underlying algorithm because of its large potential in research and commercial applications [12].

Most of the CF algorithms are limited to making only single point predictions. Metrics such as MAE and RMSE [21] were proposed in the literature to measure the prediction accuracy. But accuracy of individual predictions can not be estimated using these measures, as these measures are used to predict the overall accuracy of the recommendation algorithm. Some confidence estimation algorithms have been proposed in the literature to estimate the confidence of each prediction. But none of these algorithms provide an upper bound on the error rate. In contrast, conformal predictors are able to produce confidence measures specific to each individual prediction with guaranteed error rate.

Conformal Prediction (CP) [4,5] is the framework used in machine learning (ML) to make reliable predictions with known level of significance or error probability. Moreover, CP is increasingly becoming popular due to the fact that it can be built on top of any conventional point prediction algorithms like K-NN [6], SVM [18], decision trees [19], neural networks [20] etc. The confidence measures produced by CPs are not only useful in practice, but also their accuracy is comparable to, and sometimes even better than that of their underlying algorithms. So we use CP to associate a confidence measure for each individual prediction made by our chosen underlying algorithm.

The regions produced by any CP algorithm are automatically valid. But efficiency in terms of tightness and usefulness of prediction regions depends on the nonconformity measure (NCM) used by the CP algorithm. Moreover, we can define many different NCMs for a given underlying algorithm and each of these measures defines a different CP. So determining an efficient NCM based on the underlying algorithm is one important step in CP. In this work, we define different NCMs based on the underlying algorithm and empirically demonstrates that the CP with simple NCM which is a variant of NCM used in K-NN [6] performs well from both accuracy and efficiency perspectives.

Major contributions of this paper are: 1. Adaptation of conformal prediction to Item-based collaborative filtering. 2. Define NCMs based on similarity measure used in IBCF and empirically determine the best NCM. 3. Empirically demonstrate validity and efficiency of our conformal prediction algorithm.

The rest of the paper is organized as follows: In the next section we discuss related work. In Sect. 3, we discuss general idea of conformal prediction. In Sect. 4, we describe Item-based collaborative filtering. Section 5 describes our proposed algorithm which apply conformal prediction on top of Item-based collaborative filtering and defines different NCMs based on IBCF. Section 6 details our experimental results and show the validity and efficiency of our proposed conformal prediction algorithm. Finally, Sect. 7 gives our conclusions.

2 Related Work

In this section we review existing methods proposed in the literature to estimate confidence of CF algorithms. McNee et al. [7] estimate the confidence of an item as support for the item. Mclaughlin and Herlocker [8] proposed an algorithm for UBCF, which generates belief distributions for each prediction. Although their algorithm is good at achieving good precision by making sure that more popular items are recommended, they did not demonstrate the accuracy of their algorithm. Adomavicius et al. [1] estimate the confidence based on rating variance of each item. Shani and Gunawardana [9] defines confidence as the system's trust in its recommendations. They proposed a method to estimate the confidence of recommendation algorithms, but no experiments are provided due to the broader scope of the chapter. Koren and Sill [10] formulated the problem of confidence estimation as a binary classification problem and find whether the predicted rating is within one rating level of the true rating. Although confidence is associated with each item in the recommendation list, the proposed confidence estimation algorithm is applicable only for their proposed CF algorithm. Mazurowski [3] introduced three confidence estimation algorithms based on resampling and standard deviation of predictions to predict confidence of individual predictions.

But all of the above algorithms failed to provide an upper bound on the error rate i.e., the probability of excluding the correct class label is guaranteed to be smaller than a predetermined significance level. On the other hand, our CP algorithms produce prediction regions with a bound on the probability of error. When forced to make point predictions, the confidence of a prediction is $1-$ the second largest p-value and this *second largest p-value* becomes the upper bound on the probability of error. Moreover, with CP we can control the number of erroneous predictions by varying the significance level, thus making it suitable to different kinds of applications.

3 Conformal Prediction

In this section, we introduce the general idea behind conformal prediction [4,5]. We have a training set of examples $Z = \{z_1, z_2, ..., z_l\}$. Each $z_i \in Z$ is a pair $(x_i, y_i); x_i \in \mathbb{R}^d$ is the set of attributes for i^{th} example and y_i is the class label for that example. Our only assumption in CP is that all $Z_i's$ are independently and identically distributed (i.i.d.). Given a new object our task is to predict the class label y_{l+1}. We try out all possible class labels y_j for the label y_{l+1} and append z_{l+1} (=(x_{l+1}, y_{l+1})) to Z. Then estimate the typicalness of the sequence $Z \cup z_{l+1}$ with respect to i.i.d by using p-value function. Our prediction for y_{l+1} is the set of y_j for which p-value $> \epsilon$, where ϵ is the significance level.

One way of obtaining p-value function is by considering how strange each example in our sequence is from all other examples. To measure these strangeness values we use NCM. NCM $\mathcal{A}$ is a family of functions which assigns a numerical score to each example z_i indicating how different it is from the examples in the set $\{z_1, ..., z_{i-1}, z_{i+1}, ..., z_n\}$.

NCM has to satisfy the following properties [4]:

1. Nonconformity score of an example is invariant w.r.t. permutations. i.e., for any permutation π of $1, 2..., n$

$$\begin{aligned} \mathcal{A}(z_1, z_2, ..., z_n) = (\alpha_1, \alpha_2, ..., \alpha_n) \implies \\ \mathcal{A}(z_{\pi(1)}, z_{\pi(2)}, ..., z_{\pi(n)}) = (\alpha_{\pi(1)}, \alpha_{\pi(2)}, ..., \alpha_{\pi(n)}). \end{aligned} \tag{1}$$

2. $\mathcal{A}$ is chosen such that larger the value of α_i stranger is z_i to other examples. p-value is computed by comparing α_{l+1} with all other nonconformity scores.

$$p(y_j) = \frac{\#\{i = 1, 2, ..., l+1 : \alpha_i \geq \alpha_{l+1}\}}{l+1}. \tag{2}$$

An important property of p-value is that $\forall \epsilon \in [0, 1]$ and for all probability distributions P on Z,

$$P\{\{z_1, z_2, ..., z_{l+1}\} : p(y_{l+1}) \leq \epsilon\} \leq \epsilon. \tag{3}$$

This original approach to CP is called Transductive Conformal Prediction (TCP). The p-values obtained from CP for each possible classification can be used in two different modes: *Point prediction:* for each test example, predict the classification with the highest p-value. The confidence of this prediction is $1-$ the second largest p-value and credibility is the highest p-value(credibility tells how well the new item with the assumed label conforms to the training set of items). *Region prediction:* given the $\epsilon > 0$, output the prediction as the set of all classifications whose p-value $> \epsilon$ with $1 - \epsilon$ confidence that the true label will be in this set. A method for finding $(1 - \epsilon)$ prediction set is said to be *valid* if it has atleast $1 - \epsilon$ probability of containing the true label. *Efficiency* of CP is the tightness of prediction regions it produces. The narrower (small number of labels) the prediction region the more efficient the conformal predictor is.

4 Item Based Collaborative Filtering Algorithm

In IBCF, prediction and recommendation are based on item to item similarity. The key motivation behind this scheme is that a user will more likely purchase items that are similar to items he already purchased. This can be done as follows: Assume that I is the set of all available items. For every target user u_t, first the algorithm looks into the set of items that he has rated (training set C_t) and computes how similar they are to the target item $i_t \in S_t$ (S_t set of test items for the target user u_t) using a similarity measure, and then selects k most similar items $\{i_1, i_2, ..., i_k\}$. Once the most similar items are found, the prediction is then computed by taking weighted average of the target user's ratings on these similar items. So two main tasks in IBCF are: computing item similarities and rating prediction.

In order to apply CP to IBCF, it is appropriate to convert all ratings to binary i.e., the user likes or dislikes the item. The reason for this is twofold: first, uneven distribution of ratings in the data sets: For instance, more than 80 % of all ratings in MovieLens 100 K are greater than 2 and nearly 70 % of all ratings in Eachmovie are greater than 3. As a result, it becomes very difficult to identify k most similar items consumed by the target user which are rated as 1 when the target item rating is assumed as 1. Second, in [14] authors have shown that user's rating as the noisy evidence of user's true rating. Therefore identifying k most similar items which are rated as 1 when the new item rating is assumed as 1 does not make sense.

The simple way to convert all ratings to binary is as follows: take the middle of the rating scale as the threshold (for instance, 3 in the rating scale of 1–6) and assume all ratings greater than the threshold as liked and all other ratings as disliked. But this approach works fine when the distribution of all ratings is even which is not the case in most of the data sets. Other approach to convert the ratings into binary is, compute the average rating for every user and consider all ratings whose rating is greater than the average as liked and the all ratings below this average as disliked. This is the best approach to deal with all types of users including pessimistic, optimistic and strict users. For example, pessimistic (optimistic) users who usually give low (high) rating to every item they consume, we assume that they like items rated above their average rating and dislike items rated below the average. Similarly, in case of strict users who rate every item correctly according to whether they like that item or not (gives high rating when they like and low rating when they do not like) in which case we assume all ratings above the average (approximately equal to the middle of the rating scale) as like and other ratings as dislike [13]. This ensures that our CP algorithm have a reasonable number of liked and disliked items in the data set which makes it easier to find k similar items when the target item is assumed as like or dislike. Item similarity computation and prediction are done as follows:

– Item Similarity Computation: In our IBCF we have chosen cosine based similarity measure to compute similarities among the items. Since, we do not have rating values, our algorithm uses binary cosine similarity measure [15] that finds the number of common users between the two items i and j and is defined as follows:

$$similarity(i,j) = \frac{\#common\_users(i,j)}{\sqrt{\#users(i)}.\sqrt{\#users(j)}}. \quad (4)$$

The above equation will give the value in between [0,1]. $Similarity(i,j) = 1$ when these two items are rated by exactly same set of users. For simplicity, we use this simple cosine similarity measure as our aim is not to improve the accuracy of the algorithm, but to provide confidence to the predictions generated by the algorithm. We can use efficient similarity measures instead of this simple measure to obtain more accurate results.

– Predicting the label (like or dislike): Once the similarities are computed, find the k most similar items (k nearest neighbors) of the target item among the consumed items of the target user. Then predict the label for the target item as the most common label among its k nearest neighbors.

5 Application of TCP to IBCF

In this section we discuss how to build TCP on top of IBCF algorithm (a variant of the algorithm in [6]). We first discuss the algorithm setup and then define different NCMs based on IBCF algorithm. Finally, we present our TCP algorithm with IBCF as an underlying algorithm.

5.1 Algorithm Setup

In order to apply CP, we need a training set of examples $\{z_1, z_2, ..., z_l\}$ and a test example z_{l+1} for which we want to make the prediction. Here we discuss how this is formulated in the context of IBCF. For every target user u_t, there is a set of items W_t rated by this user and we consider a part of W_t as the training set C_t. For every item $i \in C_t$, user has assigned a label which tells whether the user liked($+1$) or disliked(-1) the item i. Therefore, $Y = \{+1, -1\}$. We also have a test set of items S_t ($W_t - C_t$) for which we hide the actual labels assigned by the user. Now, our task is to assign a label (which makes the current test item conforms to the training set) to each of the test set items with an associated confidence measure which is valid according to Eq. (3).

5.2 Nonconformity Measures (NCMs)

We propose different NCMs based on IBCF. First we introduce the terminology to define NCMs. Since we are having only two labels, $Y = \{+1, -1\}$, we are assuming that if $y = 1$ $\bar{y} = -1$ and vice-versa. Assume that $similarity_i^y$ is vector which is a sorted sequence (in descending order) of similarity of an item i with items $\in C_t$ with the same label y and $similarity_i^{\bar{y}}$ is a sorted sequence (in descending order) of similarity of an item i with items $\in C_t$ with the label $\bar{y}$. The weight for the item i with label y, w_i^y is defined as the sum of similarity values of k most similar items with the label y among the set of rated items C_t of the target user u_t. Similarly $w_i^{\bar{y}}$ is defined as the sum of similarity values of k most similar items with the label $\bar{y}$ among the items in C_t of the target user u_t.

$$w_i^y = \sum_{j=1}^{k} similarity(i,j)^y. \quad (5) \qquad w_i^{\bar{y}} = \sum_{j=1}^{k} similarity(i,j)^{\bar{y}}. \quad (6)$$

where k is the number of most similar items, $similarity(i,j)^y$ is jth most similar item in $similarity_i^y$, y is the label of the item i, $similarity(i,j)^{\bar{y}}$ is jth most similar item in $similarity_i^{\bar{y}}$, $\bar{y}$ is the label other than the label of item i.

In what follows we define NCMs based on IBCF:

$$NCM1 = k - w_i^y. \quad (7) \qquad NCM2 = \frac{1}{w_i^y}. \quad (8) \qquad NCM3 = \frac{w_i^{\bar{y}}}{w_i^y}. \quad (9)$$

1. NCM1 & NCM2: The simple NCMs for an item i are as follows: The maximum value of the similarity function defined in Eq. (4) is 1. As a result, the maximum value of w_i^y becomes k. The higher the value of w_i^y, the more conforming the item i is with respect to the other items. NCM1 will be high when w_i^y is small and smaller value of w_i^y indicates that the item with label y is nonconforming with other items of the same label. As a consequence, the higher the value of NCM1, the stranger the item i is with respect to the other examples with the same label according to the second property of NCM. Similarly, smaller the value of w_i^y the higher the value of NCM2. The higher the value of NCM2, the more nonconforming the item i is with respect to the other examples.
2. NCM3: A more efficient and a variant of NCM proposed in [6] can be defined by taking into consideration $w_i^{\bar{y}}$ along with w_i^y in computing the nonconformity score. According to NCM3, example i with label y is nonconforming when it is very similar to the items with label $\bar{y}$ (high value of $w_i^{\bar{y}}$) and dissimilar to the items with label y (low value of w_i^y).

5.3 Item-Based Collaborative Filtering with TCP (IBCFTCP)

Algorithm 1 describes the application of TCP to IBCF in detail. For every item i in S_t of the target user u_t, try all possible labels in Y and compute the typicalness of the sequence E resulting from appending i with the assumed label to C_t using p-value function which in turn uses nonconformity values (calculated using any of the NCMs discussed above) of all items in E. For region predictions, output the prediction as the set of all labels whose p-values are $> \epsilon$ with confidence $1-\epsilon$ or in case of point predictions, output the label with the highest p-value with confidence $1-$ the second highest p-value and credibility as the highest p-value.

6 Experimental Results

We tested our algorithm on four data sets: MovieLens 100K, MovieLens 1M, MovieLens-latest-small and EachMovie. We randomly selected 50 users and for each user first 60 % of the data is considered as training set and remaining 40 % is taken as the test set. Details of data sets is given in Table 1. As TCP is a time consuming approach and it increases with the number of items when applied to IBCF, we do not conduct our experiments on data sets in other domains such as books and music where there are large number of items. Moreover, we do not compare our results with other state-of-the-art algorithms, as our aim is not to improve the performance of the algorithm but to associate confidence to the predictions made by the algorithm without compromising the performance.

Algorithm 1. Item-based Collaborative Filtering with TCP

```
Input: $k$,$u_t$,$C_t$,$S_t$,$I$,$Y$,$\epsilon$
for each $i \in I$ do
  for each $j \in I$ do
    Compute the similarity between $i$ and $j$ using Eq. (4)
  end for
end for
for each $i \in C_t$ do
  Compute $w_i^y$ and $w_i^{\bar{y}}$
  Compute nonconformity scores using any of the NCMs discussed above.
end for
for each $i \in S_t$ do
  for each $y \in Y$ do
    Update the similarity values of item $i$ with other items
    Compute $w_i^y$ and $w_i^{\bar{y}}$
    Compute nonconformity score of $i$ using any of the NCMs discussed above.
    for each $j \in C_t$ do
      if label(j) = y then
        if $similarity(i,j)^y > similarity(j,k)^y$ then
          recompute the nonconformity score of item $j$
        end if
      else if $label(j) = \bar{y}$ then
        if $similarity(i,j)^{\bar{y}} > similarity(j,k)^{\bar{y}}$ then
          recompute the nonconformity score of item $j$
        end if
      end if
    end for
    Compute the p-value(p(y)) of item $i$ with label $y$
  end for
  prediction region = $\{y|p(y) > \epsilon\}$ with confidence 1-$\epsilon$
  OR
  Assign the label $y$ to item $i$ such that max{p(+1),p(-1)} = p(y)
  confidence = 1 - second highest p-value
  credibility = highest p-value
end for
```

Here we compare performance of CP with the underlying algorithm (IBCF) in terms of percentage of correct classifications (%CC) for different data sets and for different k values. In our experiments, we considered 4 different k values: 5,10,15 and 20. In order to do this comparison, we have to make single point predictions, since the underlying algorithm make only single point predictions. In single point predictions we output a label with the highest p-value. In case, if both labels share this p-value then we can take any one of these labels randomly. In our experiments we take both labels: one that is same as the true label (conforming one) and other one which is other than the true label (nonconforming one) and compute the performance of CP algorithm separately for both cases. In this way we can measure the certainty in predictions. In Table 2 we compare

Table 1. Summary of data sets

Dataset	#Users	#Items	#Ratings
MovieLens 100K	943	1682	100000
MovieLens 1M	6040	3952	1000209
MovieLens-latest-small	718	8915	100234
EachMovie	36656	1621	279983

the performance in terms of %CC of IBCFTCP with that of IBCF with both conforming labels (CL) and nonconforming labels (NL). We got same results for both NCMs1&2 in terms of %CC, validity and efficiency. So we do not show their results separately. We also calculate the uncertainty in the predictions as the percentage of items having more than one label (in our case both labels) shares the highest p-value. As the percentage of uncertainty is increased the deviation between performance of CP with conforming label and nonconforming label will also be increased as shown in Table 2.

Table 2. Performance comparison of IBCF with IBCFTCP with CLs and NLs

Dataset	Number of nearest neighbors	Algorithm								
		IBCF			IBCFTCP					
					NCMs1&2			NCM3		
		%CC		Uncertainty	%CC		Uncertainty	%CC		Uncertainty
		CL	NL		CL	NL		CL	NL	
MovieLens 100 K	5	0.668	0.668	0	0.6964	0.6492	0.0472	**0.6764**	**0.6708**	0.0056
	10	0.7178	0.6086	0.1092	0.7002	0.652	0.0481	0.6793	**0.6715**	**0.0077**
	15	0.6654	0.6654	0	0.6917	0.6494	0.0423	**0.6741**	**0.668**	0.0061
	20	0.6861	0.6232	0.0629	0.6868	0.6492	0.0376	0.6715	**0.6668**	**0.0047**
MovieLens 1M	5	0.6899	0.6899	0	0.7226	0.675	0.0476	**0.6993**	**0.6944**	0.0049
	10	0.7469	0.6317	0.1152	0.7181	0.6739	0.0442	0.7002	**0.6948**	**0.0055**
	15	0.6903	0.6903	0	0.7189	0.6746	0.0442	**0.7014**	**0.6967**	0.0047
	20	0.7269	0.6592	0.0677	0.7179	0.6739	0.044	0.6989	**0.694**	**0.0049**
MovieLens-latest-small	5	0.612	0.612	0	0.6421	0.5929	0.0492	**0.6246**	**0.6113**	0.0133
	10	0.6948	0.5388	0.1559	0.6494	0.6026	0.0468	0.6353	**0.6244**	**0.0109**
	15	0.6148	0.6148	0	0.6462	0.6122	0.034	**0.6338**	**0.6265**	0.0073
	20	0.6678	0.5636	0.1042	0.6441	0.6092	0.0349	0.6314	**0.6235**	**0.0079**
EachMovie	5	0.6798	0.6798	0	0.732	0.6344	0.0976	**0.6941**	**0.6842**	0.0099
	10	0.7454	0.6114	0.134	0.723	0.6376	0.0855	0.6904	**0.6839**	**0.0065**
	15	0.6758	0.6758	0	0.7146	0.6419	0.0727	**0.686**	**0.6776**	0.0084
	20	0.7053	0.6242	0.0811	0.701	0.645	0.056	0.6792	**0.6742**	**0.005**

From Table 2 when CLs are considered, the CP algorithm (for all NCMs) is outperforming IBCF for odd k values (5 & 15) due to its slightly higher uncertainty values compared to IBCF, whereas IBCF is performing better for even k values (10 & 20) because of its significantly higher uncertainty values.

In case of NLs, CP with NCMs1&2 is showing performance improvement for even k values and lower performance for odd k values (5 &15) compared to IBCF. On the other hand, CP with NCM3 is outperforming IBCF for all k values when NLs are taken into account. From Table 2 we can say that NCM3 is the best NCM (shown in bold numbers) as it is showing good performance with less uncertainty compared to NCMs1&2. Although NCMs1&2 are outperforming NCM3 in terms of %CC when CLs are taken into consideration, these are not good NCMs, as this improvement is due to uncertain predictions produced by these NCMs and though IBCF with CLs is performing well for even values of k compared to CP with NCM3, this is also because of high uncertainty involved in predictions made by IBCF. Uncertainty of IBCF for odd k values is 0 (because in this case there is no possibility of obtaining equal number of $+1$ s and -1 s) and for even k values its uncertainty is even greater than IBCFTCP with NCMs1&2.

All CPs are automatically valid. Notice that in IBCFTCP the training set and test set of items differ from user to user in contrast to CP in ML where the training set and test sets are fixed for the whole algorithm. So, it is necessary to show that the validity is satisfied for each user. The validity of CP with NCM3 for each user for different data sets and for $k = 5$ is shown in Fig. 1 (We got similar results for validity and efficiency for other k values. Because of space limitations we are not showing their results). Although validity for each user is also satisfied for NCMs1&2 we did not show the results here because NCM3 is the best NCM in terms of prediction accuracy as shown in Table 2 and efficiency (which will be discussed in the following paragraphs). From Fig. 2 we can see that the error values (validity) of most of the users are within the bounds of ϵ.

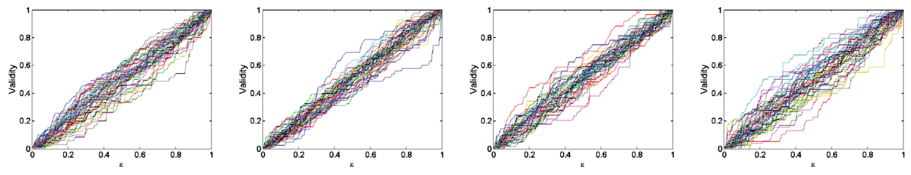

Fig. 1. Validity of IBCFTCP for different data sets (left to right:Movielens 100 K, Movielens 1M, Movielens-small-latest and Eachmovie)

Figure 2 shows average validity of IBCFTCP for different data sets and for different NCMs for $k = 5$. From Fig. 2, it is clear that prediction regions produced by CP with all NCMs are valid and follows a straight line as required.

Usefulness of CPs depend on its efficiency. Vovk et al. [11] proposed ten different ways of measuring the efficiency of CPs. In our experiments we use Observed Excess criterion which gives the the average number of false labels (ANFL) included in the prediction set at significance level ϵ. The formula to calculate ANFL is:

$$ANFL = \frac{\sum_{i=1}^{|S_t|} |\Gamma_i^{\epsilon} \setminus T_i|}{|S_t|}. \tag{10}$$

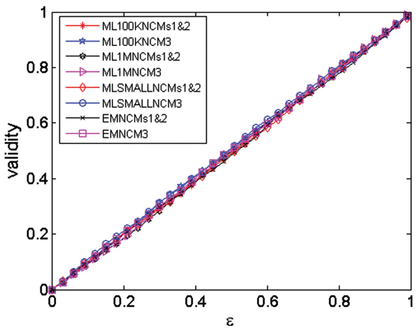

Fig. 2. Validity

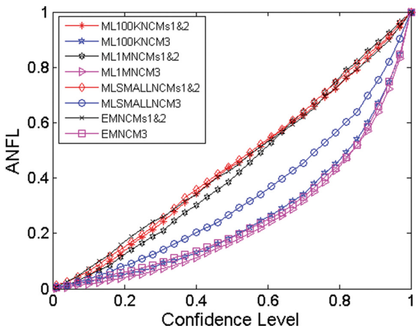

Fig. 3. ANFL

where Γ_i^ϵ is the prediction region for the i^{th} test item at significance level ϵ, T_i is the true label of the i^{th} test element, $|S_t|$ is the total number of test items.

The efficiency of CP algorithm in terms of ANFL for different data sets, different NCMs and for $k = 5$ is shown in Fig. 3 shows that CP with NCM3 is producing less number of false labels compared to NCMs1&2. From Fig. 3, we can observe that the number of false labels are decreasing with the decrease of confidence level (if we go from higher to lower confidence levels).

In addition to ANFL we use three other criteria to compute the efficiency of CP: 1. % of test elements having prediction regions with single label. 2. % of test elements having prediction regions with more than one label. 3. % of test elements having empty prediction region. A CP is said to be efficient when the % of second and third criteria are relatively small whereas the % of first should be high especially at higher confidence levels (50–99%). Our algorithm is optimal in the sense that it produces 0 % empty prediction regions from 70–99% confidence levels and $<$ 20 % from 50–70 % confidence levels, while showing moderate performance in minimizing the second one and maximizing the first one at these confidence levels. The % of test elements having prediction regions producing single labels at different confidence levels for different data sets, different NCMs and for $k = 5$ is shown in Fig. 4. From Fig. 4 we can observe that NCM3 is outperforming NCMs1&2, in producing the higher number of prediction regions with single labels. The % of single labels produced by NCMs1&2 never exceed 20 % at any confidence level, whereas NCM3 is producing sufficiently large % of single labels especially from 30 %–90 % confidence levels. Figure 5 shows the % of correct predictions among the % of single labels produced by NCM3 at each confidence level. From Fig. 5 we can see that the % of correct predictions at any confidence level is $>$ 60 %. The % of test elements having prediction regions with more than one rating at different confidence levels is shown in Fig. 6. In this case also NCM3 is performing better than NCM1&2 as NCM3 is producing less number of prediction regions with multiple labels compared to NCMs1&2 at all confidence levels and in case of NCM3 this number is zero from 0–60 % confidence levels. The % of test elements having empty prediction regions at different confidence levels is shown in Fig. 7. In this case also NCM3 is giving good results compared to NCMs1&2 by producing less number of empty prediction regions compared to NCMs1&2 and this number is zero from 70–99% confidence levels.

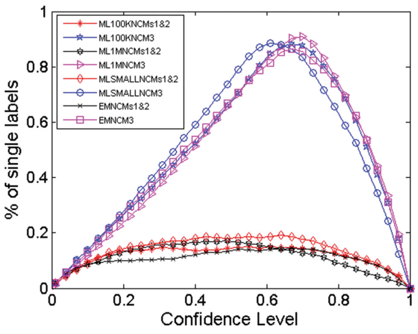

Fig. 4. % of single labels

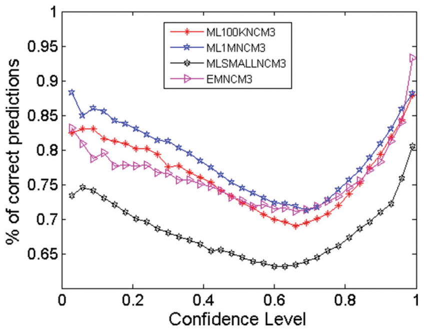

Fig. 5. % of correct predictions

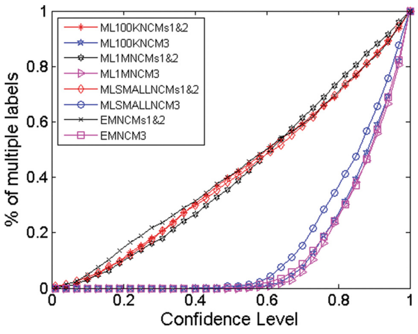

Fig. 6. % of multiple labels

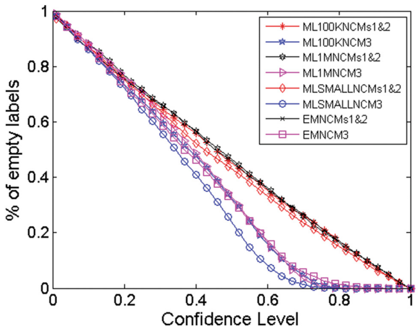

Fig. 7. % of empty labels

The mean confidence and mean credibility of single point predictions produced by IBCFTCP is shown in Table 3. We also calculated mean difference between highest p-value and lowest p-value for all test ratings. We will get confidence predictions when this difference is high. From Table 3, we can observe that this difference for NCM3 is around 50 %, whereas it is only around 15 % with NCMs1&2. Also, the mean confidence and mean credibility values produced by CP with NCM3(shown in bold numbers) are better than that of NCMs1&2.

In summary, NCM3 is the best NCM compared to NCMs1&2 in terms of prediction accuracy and efficiency. Moreover, when restricted to make single point predictions, mean confidence and credibility values produced by CP with NCM3 are higher than that of NCMs1&2.

Table 3. Mean confidence and mean credibility of IBCFTCP

Number of nearest neighbors	Performance measure	Dataset							
		MovieLens 100K		MovieLens 1M		MovieLens-latest-small		EachMovie	
		NCMs 1&2	NCM3	NCMs 1&2	NCM3	NCMs 1&2	NCM3	NCMs 1&2	NCM3
5	mean confidence	0.5841	**0.8643**	0.5864	**0.8718**	0.5881	**0.8346**	0.5738	**0.865**
	mean p1 - p2	0.1183	**0.4949**	0.1144	**0.4933**	0.1379	**0.4965**	0.1064	**0.4942**
	mean credibility	0.5347	**0.6306**	0.5284	**0.6216**	0.5503	**0.662**	0.5335	**0.6293**
10	mean confidence	0.5902	**0.8663**	0.5903	**0.8761**	0.5907	**0.8397**	0.585	**0.8681**
	mean p1 - p2	0.125	**0.4905**	0.1191	**0.4933**	0.1392	**0.4982**	0.1218	**0.496**
	mean credibility	0.5353	**0.6242**	0.5292	**0.6172**	0.5489	**0.6586**	0.5377	**0.6279**
15	mean confidence	0.598	**0.8666**	0.5947	**0.8782**	0.595	**0.8446**	0.5939	**0.8666**
	mean p1 - p2	0.1352	**0.4896**	0.1256	**0.4932**	0.1443	**0.5012**	0.1357	**0.4977**
	mean credibility	0.5376	**0.623**	0.5313	**0.615**	0.5496	**0.6567**	0.5425	**0.6311**
20	mean confidence	0.6063	**0.8675**	0.5994	**0.8783**	0.6	**0.8468**	0.6021	**0.8662**
	mean p1 - p2	0.147	**0.4927**	0.1327	**0.4931**	0.1505	**0.5017**	0.1499	**0.5003**
	mean credibility	0.541	**0.6253**	0.5337	**0.6148**	0.5509	**0.655**	0.5484	**0.6341**

7 Conclusions

In this work, we show the adaptation of CP to IBCF and proposed different NCMs for CP based on the underlying algorithm. Using our CP algorithm we are associating confidence values to each prediction along with the guaranteed error rate unlike IBCF which produces only bare predictions. Our algorithm is tested on different data sets and we experimentally proved that NCM3 is the best NCM compared to NCMs1&2 in achieving the prediction accuracy as good as the underlying algorithm with little uncertainty and also in producing efficient prediction regions. When making single point predictions, the mean confidence and credibility values produced by the proposed algorithm are reasonably high. Although our algorithm failed in producing large percentage of single labels at 90–99 % confidence levels (desired confidence levels in medical applications) we can use this algorithm to make predictions in certain recommendation domains such as movies, books, news articles, restaurants, music and in tourism where the confidence level of 50 %–90 % is acceptable.

References

1. Adomavicius, G., Kamireddy, S., Kwon, Y.: Towards more confident recommendations:improving recommender systems using filtering approach based onrating variance. In: Proceedings of (WITS 2007) (2007)
2. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of WWW, pp. 285–295 (2001)

3. Mazurowski, M.A.: Estimating confidence of individual rating predictions in collaborative filtering recommender systems. Expert Syst. Appl. **40**(10), 3847–3857 (2013)
4. Shafer, G., Vovk, V.: A tutorial on conformal prediction. J. Mach. Learn. Res. **9**, 371–421 (2008)
5. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, New York (2005)
6. Proedrou, K., Nouretdinov, I., Vovk, V., Gammerman, A.J.: Transductive confidence machines for pattern recognition. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, p. 381. Springer, Heidelberg (2002)
7. McNee, S.M., Lam, S.K., Guetzlaff, C., Konstan, J.A., Riedl, J.: Confidence displays and training in recommender systems. In: International Conference on Human-Computer Interaction (2003)
8. McLaughlin, M.R., Herlocker, J.L.: A collaborative filtering algorithm and evaluation metric that accurately model the user experience. SIGIR **2004**, 329–336 (2004)
9. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer, New York (2011)
10. Koren, Y., Sill, J.: Ordrec: an ordinal model for predicting personalized item rating distributions. In: RecSys, pp. 117–124 (2011)
11. Vovk, V., Fedorova, V., Nouretdinov, I., Gammerman, A.: Criteria of Efficiency for Conformal Prediction. Technical report (2014)
12. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
13. Lemire, D., Maclachlan, A.: Slope One Predictors for Online Rating-Based Collaborative Filtering. CoRR abs/cs/0702144 (2007)
14. Hill, W., Stead, L., Rosenstein, M., Furnas, G.W.: Recommending and evaluating choices in a virtual community of use. In: Proceedings of ACM CHI95 Conference on Human Factors in Computing Systems, pp. 194–201 (1995)
15. Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. J. Mach. Learn. Res. **10**, 2935–2962 (2009)
16. Johansson, U., Bostrom, H., Lofstrom, T.: Conformal prediction using decision-trees. In: IEEE 13th International Conference on Data Mining, pp. 330–339 (2013)
17. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. **42**(8), 30–37 (2009)
18. Saunders, C., Gammerman, A., Vovk, V.: Transduction with confidence and credibility. In: Proceedings of IJCAI 1999, pp. 722–726 (1999)
19. Johansson, U., Bostrom, H., Lofstrom, T.: Conformal prediction using decision trees. In: IEEE 13th International Conference on Data Mining, pp. 330–339 (2013)
20. Papadopoulos, H., Vovk, V., Gammerman, A.: Conformal prediction with neural networks. In: 19th IEEE ICTAI 2007, pp. 388–395 (2007)
21. Breese, J. S., Heckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on UAI 1998, pp. 43–52 (1998)