

The k Closest Resemblance Classifier for Amazon Products Recommender System

Nabil Belacel^{1,2}^a, Guangze Wei² and Yassine Bouslimani²^b

¹Digital Technologies Research Centre, National Research Center, Ottawa, Ontario, Canada

²Department of Electrical Engineering, Moncton University, New Brunswick, Canada


Keywords: Machine Learning, Supervised Learning, Classifier, Information Retrieval, Content Based Filtering, Recommender System.


Abstract: This paper presents the application of classification method based on outranking approach to Content Based Filtering (CBF) recommendation system. CBF intends to recommend items similar to those a given user would have liked in the past by first extracting traditional content features such as keywords and then predicts user preferences. Therefore content based filtering system recommends an item to a user based upon a description of the item and a profile of the user's interests. Typically, to represent user's and items' profiles the existing CBF recommendation systems use the vector space model with basic term frequency and inverse document frequency (*tfidf*) weighting. The *tfidf* and cosine similarity techniques are able, in some cases, to obtain good performances, however, they do not handle imprecision of features' scores and they allow the compensation between features which will lead to bad results. This paper introduces k Closest resemblance classifier for CBF. The detailed models in this paper were tested and compared with the well-known *tfidf* based the k Nearest Neighbor classifier using Amazon fine food and book reviews data-set. The preliminary results show that our proposed model can substantially improve personalized recommendation of items described with short text like products description and customers' review.

1 INTRODUCTION

Personalized recommendation systems have been regarded as an important mechanism to overcome the difficulty of information overload. They have become an important research area for both academia and industry. Based on the historical records of users, the recommendation system will initially recommend the information to them for choosing according to their preferences. In general, recommendation systems use mainly three type of filtering techniques: the content based filtering (CBF), the collaborative filtering (CF) and hybrid filtering (the combination of CBF and CF) (Adomavicius and Tuzhilin, 2005). The CBF technique recommends specific items that are similar to those have been already positively rated in the past by the active user. It uses only the content of items in order to make a recommendation (Lops et al., 2011). The CF system recommends items that are preferred in the past by similar users to the active user. So, CF

techniques make the assumption that the active users will be interested in items that users similar to them have rated highly. The hybrid based filtering techniques recommend items by combining CF and CBF (Adomavicius and Tuzhilin, 2005). In this paper we will focus on the CBF that compares the user's profile to some reference characteristics to predict whether the user would be interested in unseen items or not. It uses textual analysis to generate the user's and items' profile. The CBF made recommendation based on the user interests profile using features (keywords) extracted from the content of items previously rated by that user (Lops et al., 2011). To recommend items to user, the CBF follows three main functions: - updating the user profile; -2- filtering the available items with user's profiles; 3- recommending the items that better fit the profile (Castro et al., 2014). The improved method for CBF will be discussed and compared with the traditional methods. The proposed approach is illustrated using the case study of a recommender system for amazon products' description and customers' review.

^a <https://orcid.org/0000-0003-1731-3225>

^b <https://orcid.org/0000-0003-2894-5113>

2 RELATED WORK

In CF methods, recommendation is based on other users' preferences with the assumption that similar customers are more likely to have interest in same products. Thus the similarity between different customers is calculated based on their purchasing, viewing or any relevant usage history, but the indigenous properties of each product are not considered. By contrast, CBF use information about an item itself to make suggestions, and most often, contents are in plain text form. CF and CBF both have their pros and cons. CBF have the ability to recommend new items even if there is no rating provided by other users. Or in the case where the user does not want to share her/his information with other users for privacy and security reasons. The CBF technique has the advantage in giving explanations on the recommendation results. In meantime, it suffers from several difficulties as presented in the paper (Adomavicius and Tuzhilin, 2005). CBF techniques depend only on items' features. So, they require rich description of items before recommendation can be made. Consequently, the effectiveness of CBF depends on the availability of descriptive data. In the case we do not have enough user community with their rating history and we need to use some attributes of products' reviews and description rather than using only the products' ratings, we are facing CBF.

Customer reviews, opinions and shared experience are always important for marketing uses, and sometimes take a vital position in enterprises' decision making. A good understanding of customers can help enterprises better grasp the market and generate more profits while omitting customers' useful feedback means missing out the opportunity. However, despite the importance and value of such information, there is lack of comprehensive mechanism that formalizes the opinions selection and retrieval process and the utilization of retrieved opinions due to the difficulty of extracting information from text data (Aciar et al., 2006; Li et al., 2017). In this work we are focusing only on the CBF system that is based on the content of the items. To recommend items, CBF first builds the user interest profile by using contents and description of items previously rated by this user and then it filters out the recommended items that would fit the user preferences and interests.

The outline of CBF approach, as presented in Fig.1, consists of three steps (Lops et al., 2011):

1. *Content analyzer* In this step the relevant information is extracted from the text describing the items (product description). The items' documents are presented by list of keywords or terms. To extract

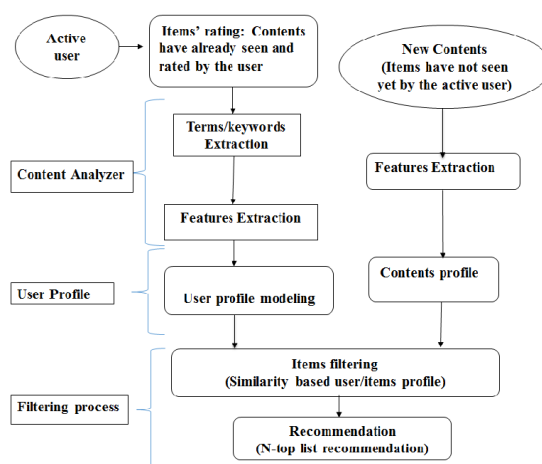


Figure 1: A general framework of CBF.

the features from the text some pre-processing steps are needed. First, We reduce the text to lower case character and removing all types of punctuation; secondly, tokens extraction or tokenization, where tokens are maximal sequence of non-blank characters. Third, we consider word stems as index terms (Tokens stemming) using Porter stemmer (Porter, 1980); and then removing the common English stop words (words that occur very often and are not relevant for discrimination). At the end a feature selection will be applied to select from the original term set (a set containing all the terms from items' documents); a subset such that only the most representative terms are used. The obtained subset will be used to generate items' profiles where each item will be presented by a set of important terms.

2. *User Profile* From the items' profiles generated in the previous phase, the user profile is constructed by taking into account the items that user liked or disliked in the past. In this phase, many machine learning algorithms can be used to learn the user profile (Lops et al., 2011).
3. *Filtering process*
Then, the system searches for relevant items by comparing the user profile with contents of the query items. And then, the system recommends items to the user that are the most similar to the items of the class representing the liked items of that user, the class "LIKE".

In the next section we will present a most used CBF system that uses models from information retrieval (Belkin and Croft, 1992) known as Keyword based Vector Space Model. We will use this approach as baseline in the experiments in order to compare with our proposed method.

The CBF using Vector Space Modeling (VSM) follows the same process of text classification by categorizing documents into predefined thematic categories. This is often uses the supervised learning and conducted in two main phases: the document indexing and classifier learning (Sebastiani, 2002): In document indexing the numeric representation of document is created by applying the two steps on documents: first a subset of terms from all terms occurring in the whole collection is selected and then term weighting is calculated by assigning a numeric value to each term in order to build the profiles of documents based on its contribution to each document. In the classifier learning a document classifier is developed by learning from the numeric representations of the documents.

In information retrieval and VSM, term weighting is formulated as *term frequency.inverse documents frequency* known as *tfidf*. The *tfidf* is one of the most popular term-weighting techniques for CBF (Lops et al., 2011). For instance, 83% of content-based recommender systems in the domain of digital libraries use *tfidf* (Beel et al., 2016). CBF using Vector Space Model (VSM) is often conducted in three phases:

1. **Feature extraction**, each product is represented by a subset of terms from all terms occurring in the items collection
2. **Term weighting**, the items' features are weighted using the most common weighting method in the VSM known as term frequency-inverse document frequency *tfidf* method. The *tf* gives a local view of term, expresses the assumption that multiple appearance of term in a document are no less important than single appearance. The *idf* gives a global view of terms across the entire collection assuming that rare terms are no less important than frequent term. For more details on *tfidf* for CBF the readers can refer to (Pazzani and Billsus, 2007; Lops et al., 2011). In *tfidf* the user profile is represented by a vector of weights where each component denotes the importance of term to user.
3. **k-nearest neighbor classifier and cosine similarity measure**, from the two above phases, the user profile and the content of new items are represented as *tfidf* vectors of terms' weight. The CBF system calculates the similarity between the documents previously seen and rated by the user and the new document. Prediction of user's interest in particular document is obtained by cosine similarity. As pointed out in the reference (Lops et al., 2011) cosine similarity is most widely used

to determine the closeness between two documents.

The *k*-Nearest Neighbor (*k*-NN) is a classical method for recommender systems (Lops et al., 2011). *k*-NN is a basic machine learning algorithm used for classification problems. It compares the new item with all stored labeled items in training set using the cosine similarity measure and determines the *k* nearest neighbors. The class label of the testing item or new item can be determined from the class labels of the nearest neighbor in the training set. Each item in the training set is presented by a weighted vector, which each component *j* presents the *tfidf* of corresponding term *t_j*. For each item in the testing we calculate the *tfidf* on the *m* terms selected from training set. The training phase of the algorithm consists only of storing the attribute vectors with their class label in memory. *k*-NN algorithm compares the all stored items to query item using a cosine similarity function and determine the *k* nearest neighbors. A majority voting rule is applied to assign a query item to a class "LIKE" or "DISLIKE". The *k*-NN classifier is one of the successful techniques for CBF.

Although *k*-NN classifier has been successfully applied to some CBF applications, it suffers from some limitations such as: it requires a high computation time because it needs to compute distance of each query item to all training items (it does not have a true training phase, all the training set is used); the pre-processing, normalization or change of input domain is often required to bring all the input data to the same absolute scale. The number *k* in the *k*-NN is given a priori. So, if one changes the number *k*, the assignment decision may be also changed. To address these issues a *k*-CR was introduced. In the following section, we describe our CBF based *k*-CR.

3 METHODOLOGY

Our methodology follows the same process of text classification by categorizing documents into predefined thematic categories. This is often uses the supervised learning and conducted in two main phases: the document indexing and classifier learning (Sebastiani, 2002): In document indexing the numeric representation of document is created by applying the two steps on documents: first a subset of terms from all terms occurring in the whole collection is selected and then term weighting is calculated by assigning a numeric value to each term in order to build the profiles of documents based on its contribution to each document. In the classifier learning a document classifier is developed by learning from the numeric represen-

tations of the documents. The outline of our proposed CBF approach consists of the three steps as presented in Fig.2):

3.1 Content Analyzer

A list of terms are extracted from the item's document. Each item is described by set of most important terms. After the pre-processing was done, the 100 more frequent words are extracted to generate the items' profiles.

3.2 User Profile/ Learning a User Model

The contents of items are converted to structured data by selecting the 100 attributes generated in content analyzer component. From the structured historical data a classification learning is applied to build user's preference model. Therefore, the training data is divided into two classes:- class "LIKE": the items that user likes; class "DISLIKE": the items that user doesn't like. In this work, we have proposed new CBF methodology based on the k -Closest Resemblance classifier.

k-Closest Resemblance classification method: The k -Closest Resemblance (k -CR) is a prototype based classification method. The k -CR method is based on the scoring function to determine a subset of prototypes representing the closest resemblance with an item to be classified (Belacel, 1999; Belacel, 2004). It applies the majority-voting rule to assign an item to a class. The scoring function is based on outranking approach developed by Roy (Roy, 2013) and following the same methodology of *PROAFTN* classifier presented in (Belacel, 2000; Belacel, 2004; Belacel and Cuperlovic-Culf, 2019). The k -CR procedure proceeds in two phases:

1. **Prototypes Learning:** For each class C^h ($h = 1$ presenting the class "LIKE" AND $h = 2$ presenting the class "DISLIKE", k -CR determines a set of prototypes or items' profiles. For class "LIKE" the prototype is representing by the vector (b^h) , $h = 1$ and for the class "DISLIKE" the prototype is representing by the vector (b^h) , $h = 2$. The profiles are considered as good representative of their class and are described by the score of the n features. The profile of the recommended items representing the class "LIKE" and not-recommended items representing the class "DISLIKE". In the learning phase we determine for each class a reference profile. The profile (b^1) representing the profile of items that the target user likes and (b^2) representing the profile of the items that the target user dislikes. To determine the profile of

each class, we use the training set that contains set of items that the user already seen and rated. More precisely for each profile and each feature of each class, an interval is determined. To define these intervals k -CR follows the same discretization approach for learning *PROAFTN* classifier described in (Belacel and Cuperlovic-Culf, 2019). Once the prototypes of the classes are built, k -CR will proceed to phase 2.

2. **Prediction/Classification:** To classify an unlabeled sample, the k -CR determines the smallest possible subset of prototypes which are closest to an item to be classified. Based on this subset of closest prototypes, the decision to classify or not an item to a class is made by applying the same majority voting rule used in k -nearest neighbor classifier. To classify an item to LIKE or DISLIKE class, k -CR applies the following steps:
 1. *Preference Relation between the Prototypes:* k -CR takes as input the partial distance and similarities induced by the the set of features and aggregates them into global preference relation $P^s(b^h, b^l)$, b^h and b^l represent two prototypes of classes C^l and C^h respectively. The preference relation P^s expresses the degree with which the resemblance between the item s and the prototype b^h is stronger than the resemblance between the item s and the prototype b^l .
 2. *Scoring Function:* Based on the preference relations P^s between the whole prototypes of classes, k -CR selects the best prototypes in terms of their distance or resemblance with the item to be classified. The scoring function is used to select a subset of prototypes, eventually reduced to one prototype, that are more closely to the item s . Different scoring functions can be used, for more details please see (Belacel, 1999; Belacel, 2004);
 3. *Assignment Decision/Class Prediction:* Once a subset of k closest resemblance prototypes presenting the items' profiles to the query item s is determined, a majority voting rule is applied to assign the item s to the closest class. More details on k -CR are given in (Belacel, 1999; Belacel, 2004; Belacel and Boulassel, 2004).

3.3 Recommendation

The k -CR procedure ranks the items in the class "LIKE" from the best to the worst. Then, based on the ranking, our CBF system selects the N -top items from the class "LIKE" to be recommended to the user.

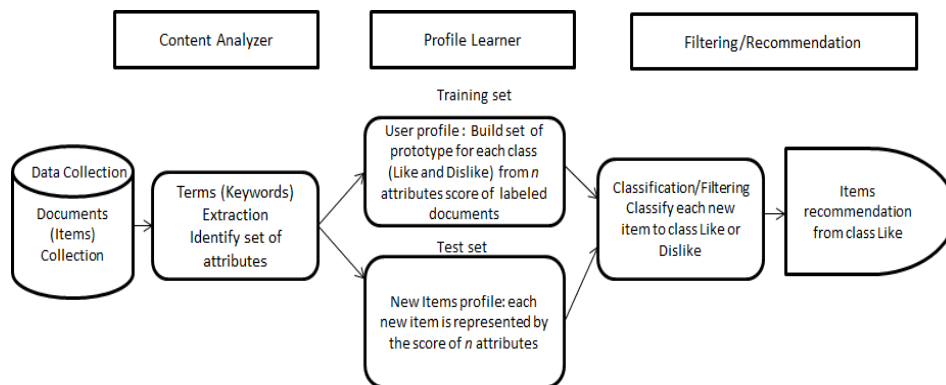


Figure 2: A general framework of our proposed CBF.

4 EXPERIMENT

4.1 Data Set

For our experiments we have used two data sets: the Amazon Fine Food and Book review. The two data sets consist of product and user information, ratings, and a short text review (McAuley and Leskovec, 2013; He and McAuley, 2016). The data sets used in our experiments are obtained from Stanford University’s Snap Dataset, (<https://snap.stanford.edu/data/web-Amazon.html>). The Amazon Fine Food data consists of 568,454 reviews between October 1999 and October 2012; 256,059 users and 74,258 items. The Amazon book data set consists of 12,886,488 reviews; 2,588,991 users and 929,264 items. The both datasets contain 8 fields: product ID, user Id, profile name, helpfulness, rating or score, time, product summary and text review. The following example is one of user’s review of dog food:

```

product/productId: B001E4KFG0
review/userId: A3SGXH7AUHU8GW
review/profileName: delmartian
review/helpfulness: 1/1
review/score: 5.0
review/time: 1303862400
review/summary: Good Quality Dog Food
review/text: "I have bought several of the Vitality
canned dog food products and have found them all
to be of good quality. The product looks more like a
stew than a processed meat and it smells better. My
Labrador is finicky and she appreciates this product
better than most".
  
```

In our experiments we have used only five attributes: <UserId, ProductId, Score/Rating, Sum-

mary, review>. The rating value is given by an integer number between 1 and 5. To build the learner profile we consider rating greater to 3 as the user likes the item otherwise the user dislike it. For the both data sets, we have selected only users that have more than 60 reviews. In total we have used about 70 users for each data set.

4.2 Evaluation and Metrics

In our experiments, we have used the 5-fold cross validation technique. For each user, the items already reviewed and rated randomly split in 5 folds. One of the fold was used for testing and the 4 remaining folds were used for training. The training set was used for building the prototypes of each class. The experiments were executed five times and the average values of the evaluation metrics were reported. The following evaluation metrics were used to evaluate the CBF approaches: precision, recall and F1-measure. These metrics are well-known in information retrieval and widely used in content based filtering to measure the effectiveness of CBF recommendation (Castro et al., 2014). Precision is the ratio of the recommended items that are actually liked by users to recommended items. Recall is the ratio of the recommended items that are actually liked by user to the total number of items that user liked in user’s test data set. F1 measure the trade-off between precision and recall ($F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$).

4.3 Results & Discussion

A comparative analysis among the two models VSM based k -NN and VSM based k -CR previously detailed has been carried out by using the precision, recall and $F1$ -score, according to the number of recommendation that the CBF provides to the user. To test our re-

sults we have used the Precision-Recall (PR) curves for all customers. These curves show the relation between the precision and recall of the two recommenders based CBF. In Fig. 3 and Fig. 4 the PR curves for VSM based k -NN and VSM based k -CR on respectively Amazon Fine Food and Amazon book data sets are plotted.

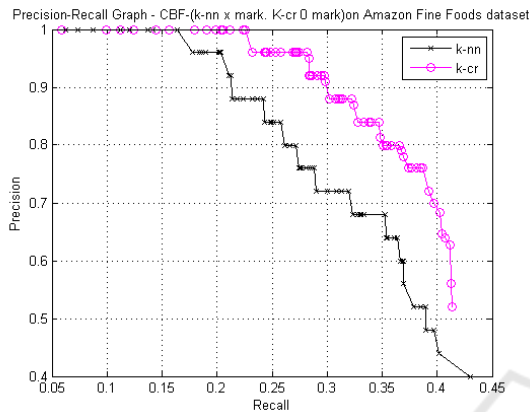


Figure 3: Precision Recall curves for the CBF based k -NN and CBF based k -CR on Amazon Fine Foods data.

As shown in fig. 3, the CBF based k -CR outperforms significantly CBF based k -NN on Amazon Fine Food. The CBF based k -CR has higher average precision for a corresponding recall than CBF based k -NN. If we consider the average recall of 40%, the CBF based k -CR has average precision about 70%, whereas the average precision of the CBF based k -NN is about 45%. As shown in fig. 3 the CBF based k -CR maintains higher precision than k -NN when the recall has values 20% or higher and has almost the same precision for recall value less than 20%. The same results for the recall, the k -CR has higher recall for a corresponding precision. For example, if we consider a precision of 70%, the CBF based k -CR has an average recall about 40%, whereas the average recall of CBF based k -NN is about 32%.

In the case of Amazon book data set the results are not obvious and there is no clear difference between the performances of k -NN and k -CR. As shown in fig. 4, when the average recall is set between 25% and 35%, the CBF based k -CR outperforms slightly the CBF based k -NN. However, for the average recall values greater than 0.38% the CBF based k -NN is better than k -CR.

The reason why the Amazon book data is less obvious than other data set may lie to the difference between the both data sets used in our experiments. For example the most customers selected in our experiments in the book data sets have 100 % positive reviews, which will have problem to build user profile

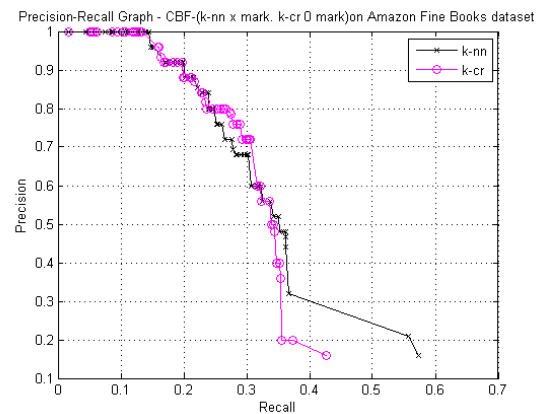


Figure 4: Precision Recall curves for the CBF based k -NN and CBF based k -CR on Amazon books data.

for the class C^2 representing the items that the user does not like.

Since the precision and recall curves do not give more detail results such as how many times the recommender A is better than recommender B according to the average precision per user. Hence, the results of the both recommenders are divided into three classes: better, equal and worse performances as presented in Fig. 5 and Fig. 6.

In the Fig. 5 and Fig. 6 all users' measurements are sorted by review numbers. Dark color represents better and lighter represents worse performance according to precision. As shown in both figures the CBF based k -CR yield better results than CBF based k -NN. For the Amazon Food data 71% of customers the k -CR has better precision than k -NN and only in 11% of users k -NN is better than k -CR. On the other hand, in 51% of users the k -CR has better precision than k -NN and only in 27%, it is better than k -CR. In this work, to learn k -CR classifier we have considered only one profile per class. We think by adding more profiles per class we can improve the performances regarding the precision as well as the recall.

5 CONCLUSIONS

In this work we have applied learning user models based on the classifier k -CR for CBF using only the short descriptions and the reviews of the products. The system can recommend products to user based only on the content of his past reviews with the reviews of other customers on the products in Amazon database. Currently, we are working in improving our CBF by incorporating other features and by combining k -CR with other machine learning like deep learning and support vector machine to solve very large



Figure 5: A comparison results for all users for food data. Red color represents the k -CR performances whereas the grey color represents the k -NN results. The lighter the color, the worse is the results of k -CR. The pre-avg in the graph presenting the average precision for each user.



Figure 6: A comparison results for all users for book data. Red colors represent the k -CR performances whereas the grey color represents the k -NN results. The lighter the color, the worse is the results of k -CR. The pre-avg in the graph presenting the average precision for each user.

data sets. We are also investigating if the proposed models would perform well in various other recommendation applications based on short text, e.g. twitter, blogs and RSS feeds.

ACKNOWLEDGEMENTS

The authors thank the MITACS Globalink Internship program for undergraduate students for funding the internship of the second author.

REFERENCES

- Aciar, S., Zhang, D., Simoff, S., and Debenham, J. (2006). Recommender system based on consumer product reviews. In *Proceedings of the 2006 IEEE/WIC/ACM international Conference on Web intelligence*, pages 719–723. IEEE Computer Society.
- Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749.
- Beel, J., Gipp, B., Langer, S., and Breiting, C. (2016). paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338.
- Belacel, N. (1999). *Méthodes de classification multicritère: Méthodologie et application à l'aide au diagnostic médical*. PhD thesis, Univ. Libre de Bruxelles, Belgium, Brussels.
- Belacel, N. (2000). Multicriteria assignment method PROAFTN: methodology and medical application. *European Journal of Operational Research*, 125(1):175–183.
- Belacel, N. (2004). The k-closest resemblance approach for multiple criteria classification problems. In Hoai, L. and Tao, P., editors, *Modelling, Computation and Optimization Information and Management Sciences*, pages 525–534. Hermes Sciences Publishing.
- Belacel, N. and Boulassel, M. R. (2004). Multicriteria fuzzy classification procedure procfnt: methodology and medical application. *Fuzzy Sets and Systems*, 141(2):203–217.
- Belacel, N. and Cuperlovic-Culf, M. (2019). Proaftn classifier for feature selection with application to alzheimer metabolomics data analysis. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(11):1940013.
- Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, 35(12):29–38.
- Castro, J., Rodriguez, R. M., and Barranco, M. J. (2014). Weighting of features in content-based filtering with entropy and dependence measures. *International Journal of Computational Intelligence Systems*, 7(1):80–89.
- He, R. and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, pages 507–517. International World Wide Web Conferences Steering Committee.
- Li, Y., Jiang, J., and Liu, T. (2017). Inferring user consumption preferences from social media. *IEICE TRANSACTIONS on Information and Systems*, 100(3):537–545.
- Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer.
- McAuley, J. J. and Leskovec, J. (2013). From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, pages 897–908. International World Wide Web Conferences Steering Committee.
- Pazzani, M. J. and Billsus, D. (2007). *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter Content-Based Recommendation Systems, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Porter, M. (1980). An algorithm for suffix stripping, program 14 (3).
- Roy, B. (2013). *Multicriteria Methodology for Decision Aiding*. Nonconvex Optimization and Its Applications. Springer US.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.