**Recommendation systems predict user’s preference to suggest items. Collaborative filtering is the most popular method in implementing a recommendation system. The collaborative filtering method computes similarities between users based on each user’s known preference, and recommends the items preferred by similar users. Although the collaborative filtering method generally shows good performance, it suffers from two major problems – scarsità dei dati and scalability. In this paper, we present a model-based recommendation algorithm that uses multi-level association rules to alleviate those problems. In this algorithm, we build a model for preference prediction by using association rule mining. Multi-level association rules are used to compute preferences for items. The experimental results show that applying multi-level association rules is efficienti , and performance of the algorithm is improved compared with the collaborative filtering method in terms of the recall and the computation time.**

Recommendation sistem predice le preferenze dell'utente basandosi sui dati storici. Collaborative filtering è il metodo più utilizzato e implementato, ma soffre di scarsità dei dati e scalabilità. In questo articolo verrà utilizzato un metodo multi livello basato su regole di associazione. I risultati dell'esperimento dimostrano l'efficienza dell'algoritmo confrontato con il collaborative filtering in termini di recall e tempo di calcolo.

**Recommendation systems predict a user's preference and suggest items by analyzing the past preference information of users. The data used for recommendation can be stored as a preference matrix that represents each user's preference for each item. The preference of a user can be obtained by either explicit or implicit rating. The explicit rating means that a user provides his or her preference to an item on a numerical scale. Implicit rating means that user's actions such as purchase of an item or the click on a web page is interpreted as preferences. Items may be a simple set in one level, or may be organized into a hierarchical category structure with multi-levels, like classification of goods in a department store.**

*I dati utilizzati dall'algoritmo di reccomandation system sono salvati in matrici di preferenza e rappresentano le preferenze utente per ogni elemento. Le preferenze dell'utente possono essere ottenute sia con una valutazione implicita sia esplicita. L'esplicita significa che l'utente assegna ad ogni elemento una preferenza numerica. L'implicita sono le azioni che compie (ad esempio quello che cerca, cosa visita etc etc). Gli elementi possono essere organizzati gerarchicamente oppure in una struttura multilivello.*

**The collaborative filtering is widely used as a recommendation algorithm. It first finds users who have similar preferences with the target user, and predicts preferences of the target user based on preferences of the similar users. GroupLens[10] and Ringo[13] were the first recommendation systems using CF algorithms. GroupLens recommended Usenet news articles based on user’s ratings, and used Pearson correlation as similarity measure. Ringo recommended music by using weighted average of ratings of restricted number of similar users.**

Collaborative filtering è usato ampiamente come algoritmo in questi casi. Per prima cosa individua utenti con gli stessi gusti, basandosi sulle loro similarità. GroupLEns e Ringo furono I primi algoritmi di questo genere. GL suggeriva articoli a utenti del web, e usava la correlazione Pearson come misura di similarità. Ringo suggeriva musica valutando la media delle valutazioni di un ristretto gruppo di utenti simili.

**Also, there have been many researches for performance analysis and improvement of recommendation systems. Breese[4] compared CF with clustering model and Bayesian model. Helocker[7] tested various similarity measures including Pearson correlation, Spearman correlation, vector similarity, and entropy. He also applied various weighting schemes such as similarity weighting, significance weighting, and variance weighting. To reduce computing time, dimensionality reduction methods using SVD(singular value decomposition) were proposed by Billsus[3] and Sarwar[12]. Combination of collaborative filtering method with content-based or knowledge-based approach ware suggested by Balabanovic[1], Burke[5], and Nguyen[9]. Lin[8] and Sarwar[11] applied association rule mining technique to collaborative recommender system.**

Tuttavia l'almito di questa ricerca era molto interessante per I ricercatori. Breese comparò modello CF con clustering e modello Bayesiano. Helocker testò numerose misure di similarità includendo Pearson, Spearman, vettori di similarità e entropia. Applicò anche vari schemi di pesatura e varianza. Per ridurre il tempo della Billus e Sarwar proposero un metodo di riduzione di dimensione usando SVD (singular value decomposition). Una combinazione di CF con un approccio content-base o knowledge-base fu proposto da Balabanovic, Burke, e Nguyen. Lin e Sarwar applicarono tecniche di mining con regole di associazione al collaborative reccomender system.

**The collaborative filtering is a *memory-based* algorithm. Although it generally shows good performance when sufficient explicit preference information is given, it suffers from two major problems – the data sparseness and the scalability. When the number of known preferences is very small – in other words, the user-item preference matrix is very sparse – performance of recommendation can be very poor, since it becomes difficult to find similar users. This is a critical problem because in practical applications, such as recommending items in a web shopping mall, the number of known preferences (purchasing an item for example) is very small compared to the total number of items. Also, due to the similarity computation between users, computation time for recommendation increases as the number of users grows. This scalability problem may also be a critical problem for practical applications with millions of users.**

Il CF è un algoritmo memory-based. Tuttavia le migliori prestazioni le ha quando sono date sufficienti preferenze esplicite. Soffre di scarsità dei dati e scalabilità. Quando i dati sono pochi le prestazioni sono davvero scarse e diventa difficile individuare similarità tra gli utenti. Questo è un problema critico nel lato pratico. E anche sulla valutazione temporale, il tempo di calcolo aumenta al crescere degli utenti. Con milioni di utenti diventa un problema critico.

**In this paper, we present a *model-based* recommendation algorithm that uses multi-level association rules to alleviate those problems. In this algorithm, we build a model for preference prediction by using association rule mining. Multi-level association rules are used to compute preferences for items that are not covered by the single-level association rules due to the data sparseness. We compared performance of the suggested algorithm with that of the collaborative filtering algorithm and single-level association rule algorithm in terms of the accuracy and the computation time.**

in questo articolo verrà presentato un modello di reccomandation che usa regole di associazione multi-livello per alleviare questo pèroblema. E' stato costruito un modello per predire le preferenze usando associacion rules mining. Le multi-level AR sono state usate per calcolare preferenze di elementi che non sono coperti da un solo.livello AR. Infine mostreremo i risultati sia a livello di tempo che di prestazioni.

**2.1. Association rule mining**

**Association rule mining is to search for interesting relationships between items by finding items frequently appeared together in the transaction database. If item B appeared frequently when item A appeared, then an association rule is denoted as A  B (if A, then B). The *support* and *confidence* are two measures of rule interestingness that reflect usefulness and certainty of a rule respectively [6].**

Mining di AR serve per cercare interessanti relazioni tra elementi che trovano numerose relazioni nel database. Se B appare frequentemente quando appare A, allora c'è una regola di associazione denotata come A  B (if A, then B). Il supporto e la confidenza sono due misure di questa relazione.

**Support, as usefulness of a rule, describes the proportion of transactions that contain both items A and B, and confidence, as validity of a rule, describes the proportion of transactions containing item B among the transactions containing item A. The association rules that satisfy user specified minimum support threshold (*minSup*) and minimum confidence threshold (*minCon*) are called strong association rules.**

Supporto descrive la proporzione delle transazioni che contengono sia A che B e la confidenza descrivono le transazioni che contengono B quando è presente anche A. La regola di associazione che soddisfa la soglia di mimino supporto(minSup) e soglia di minima confidenza(minCon) sono chiamate regole di associazione.

**In our approach, each user’s preference is represented as a boolean vector – 1 means that the user prefers the item. Then we regard each user’s preference vector as a transaction, and search for association rules among items.**

In questo approccio ogni preferenza è rappresentata come un vettore booleano. 1 significa che l'utente preferisce l'elemento.

**2.2. Recommendation using association rules**

**Recommendation using association rules is to predict preference for item *k* when the user preferred item *i* and *j*, by adding confidence of the association rules that have *k* in the result part and *i* or *j* in the condition part. Sarwar[11] used the rule with the maximum confidence, but we used the sum of confidences of all rules in order to give more weight to the item that is associated with more rules.**

Reccomandation usando regole di associazione per predire una preferenza per l'elemento k quando un utente preferisce I e j, aggiungendo confidenza alla regola di associazione che ha k nel risultato e I e j nella condizione. Sarwar usò le regole con massima confidenza, ma qui usiamo invece la somma delle confidenze di tutte le regole in ordine, per dare più peso agli elementi che sono associati con più regole.

**Recommendation using association rules describes as follows. Let P be the preference matrix of *n* users on *m* items. In this matrix, *pij* is 1 if the user *i* has preference for the item *j*, and 0 otherwise. Let A be an association matrix containing confidence of association rules of *m* items to each other. The matrix A is computed from P. In this matrix, *aij* is confidence of association rule *i*  *j*. Then the recommendation vector r for the target user can be computed from the association matrix A and the preference vector u of the target user as equation (1). The top-N items are recommended to the target user based on the values in r.**

Descritto come segue: Sia P la matrice di preferenza di n utenti con m elementi. In questa matrice, *pij* è 1 se l'utente I ha preferenza per l'item j. 0 altrimenti. Sia A una matrice di associazione che contiene la confidenza delle regole di associazione di m elementi con gli altri. La matrice A è calcolata da P. In questa matrice aij è la confidenza della regola di associazione *i*  *j.* Quindi il vettore di reccomandation r per il target utente puà essere calcolato dalla matrice di associazione A e il vettore di preferenza u del target utente (come nella seguente equazione):

OLE-objectOLE-object

I migliori N elementi sono raccomandati come utenti targer per I valori di r.

**2.3. Recommendation using multi-level association rules**

**In recommendation using association rules, if the amount of available preference information is small, then the number of strong association rules would also be small, and consequently the association matrix becomes very sparse. Actually, in case the purchase data is used as preference of users, the number of applicable association rules is very small because each user usually purchases only small number of items. In this case, it is impossible to predict preferences for most of items and performance of recommendation becomes very poor. To overcome this problem, we present a recommendation method using additional information – the association rules between higher-level categories – if items are organized into a hierarchical category structure.**

In reccomandation usando AR, se il totale delle preferenze disponibili è piccolo, allora il numero di forti AR sarà piccolo e di conseguenza la matrice di associazione diventa molto scarsa. Attualmente gli utenti acquistano pochi elementi, e quindi è piccolo il numero di AR forti. IN questo caso è impossibile predire preferenze per molti elementi e le performance diventano decisamente scarse. Per risolvere questo problema, presentiamo un metodo di reccomandation che usa ulteriori informazioni (AR tra categorie al livello superiore) che sono organizzate in una struttura gerarchica.

**We first find the category *ci* that the preferred item belongs. If there is a category association rule of the form *ci*  *cj*, then we give certain amount of preference to all items that belong to category *cj*.**

**Recommendation using multi-level association rules describes as follows. Let Ck be a category relation matrix that represents the inclusion relation between lower level(k-1) category and higher level(k) category (level 0 category means the item). In this matrix, *Ck*(*i*,*j*) is 1 if category *i* in level k-1 belongs to category *j* in level k, and 0 otherwise. Using this matrix, the preference matrix Pk and the preference vector uk of the target user for level k category is computed as follows:**

SI individua innanzitutto la categoria ci. Se c'è una AR di categoria nella forma *ci*  *cj*, allora verranno date deppe preferenze a tutti gli elementi contenuti nella categoria cj.

Reccomandation su multi-level AR è descritto come segue. Sia Ck una matrice di relazione categoria che rappresenta la relazione di inclusione tra le categorie del livello inferiore (k-1) con quelle del livello superiore (k). In questa matrice *Ck*(*i*,*j*) è 1 se la categoria I nel livello k-1 include la categoria j nel livello k, 0 altrimenti. Usando questa matrice, la preferenza della matrice Pk e il vettore preferenza Uk del target utente per il livello k di categoria è calcolato come segue:

OLE-objectOLE-object

OLE-objectOLE-object

**In equation (2), elements of Pk for k≥1 are not binary. If a user has preference to multiple items in a category, it can be greater than 1. The association matrix Ak for level k category is computed from Pk. It consists of confidence of association rules between categories. Note that all diagonal elements of Ak should be 1 to give preference to other items in the same category the user preferred. In other words, the confidence of association rule *ci*  *ci* is always 1 for level k ≥1. From pre-computed association matrices and category relation matrices, we can compute the recommendation vector r as follows:**

nell'equazione 2, gli elementi di Pk per k>=1 non sono binari. Se un utente ha preferenze per più elementi in una categoria, può essere maggiore di 1. La matrice di associazione Ak per il livello K di categoria è calcolata da Pk. Essa consiste nella confidenza di AR tra categorie. Notare che tutti hli elementi diagonali di Ak dovrebbero essere 1 come preferenza di altr elementi della stessa categoria. IN altre parole, la confidenza della AR *ci*  *ci* è sempre 1 per ogni k>=1. Dalla matrice di associazione pre calcolata e dalla matrice di relazione delle categorie si possono calcolare I vettori di raccomandazione come segue:

**In equation (4), k is the weight to preference computing by level k association rules, and the sum of weights is 1. The weights for the best recommendation are determined empirically. Each predicted preference of all level categories is normalized to [0.5, 1] through sigmoid function respectively.**

Nell'qeuazione 4 alfaK è il peso della preferenza calcolata dalle AR del livello K, e la somma dei pesi è 1. I pesi per la migliore raccomandazione sono determinati empiricamente. Ogni preferenza predetta per tutti I livelli di categoria è normalizzata tra 0.5-1 attraverso una funzione sigmoide.

**3. Experiment and Results**

**3.1. Dataset**

**We used two separate datasets for the experiments. The MovieLens[14] is a set of ratings to movies. The users rated each movie with 0 to 5 integer values. We selected the users who rated at least 20 movies, which become a set of about 100,000 ratings from 943 users on 1682 movies. MovieLens dataset are organized into two levels – there are 18 categories (genres) as higher level of items (movies). A movie belongs to one or more genres. We assumed that the explicit ratings for items are not generally given, and so converted the original ratings into preference of binary form. To reflect user’s rating scale, each rating was converted to 1 if the rating is greater than the user’s average rating, and 0 otherwise.**

Sono stati usati due dataset separati per gli esperimenti. Movielens che è un insieme di votazioni dei film. Gli utenti votano un film con un valore tra 0 e 5. Sono stati selezionati gli utenti che hanno votato almeno 20 film, e quindi sono state selezionate in totale 100000 votazioni con 943 utenti e 1682 film. Movielens dataset è organizzato in due livelli: ci sono 18 categorie (generi) ad alto livello di elementi (film). Un film contiene uno o più generi. Assumiamo che la v otazione esplicita non sia data generalmente, e quindi si convertono le valutazioni originarie in una forma binaria. Per riferirsi ad una scala di votazione, ogni valutazione è convertita in 1 se è maggiore della media di valutazione dell'utente, 0 altrimenti.

**The KDD Cup 2000[15] domain contains clickstream and purchase data from Gazelle.com, a legwear and legcare web retailer. We selected a set of 2,295 purchase data from 793 users on 253 items. KDD dataset consists of three category levels – there are 39 items as category level 1 item, and 2 items as category level 2 item. Each level’s item belongs to one of higher level items.**

Il KDD Cup contiene il flusso di click e acquisti fa Gazelle.com. Sono stati selezionati 2295 acquisti da 793 utenti su 253 elementi. KDD consiste in 3 categorie e ci sono 39 elementi nel livello 1 e due nel livello 2. Ogni livello è contenuto in uno più alto.

**3.2. Experimental method and metric**

**We divided dataset into 80% of training set and 20% of test set, and averaged the results of 5 trials with different test set (5-fold cross validation). To measure accuracy of recommendation, we hided one preference data for each user in test set and recommended top-N items with highest predicted preference based on remaining preference information. Accuracy is computed as the ratio of correct prediction. If the hided item is actually recommended – contained within the top-N recommendation list, it’s called *hit*. We used the Recall, the overall number of hits over the total number of preferred items in test set (the number of ‘1’s in matrix P), as the accuracy measure.**

Il dataset è stato diviso per 80% in training e 20% di test e I risultati mediati su 5 trial con differenti test set (5 fold cross validation). Per misurare l'accuratezza è stata nascosta una preferenza per ogni utente nel test set e raccomandati I migliori N elementi con la predizione più alta basati sulle informazioni rimanenti. L'accuratezza è calcolata sull'aspetto delle predizioni corrette. Se l'elemento nascosto è raccomandato (contenuto nei top-N raccomandati, è chiamato hit). Viene anche calcolata la recall, cioè il numero di hit sul numero totale di elementi preferiti nel test set.

**The first experiment is the performance comparison between two different methods of applying association rules – using a rule with the maximum confidence and using the sum of confidences, when more than one association rules are available. The second experiment is the performance comparison among simple frequency method(SF), collaborative filtering method(CF), single-level association rule method(AR+), and multi-level association rule method(MAR). The simple frequency method is the basic recommendation method that recommends the most popular top-N items to all users by counting overall preferred frequencies for each item from the preference matrix.**

Il primo esperimento è la comparazione di peformance tra due metodi differenti applicando AR. Usando la max confidence e la somma delle confidence (quando una o più regole di associazione sono disponibili). Il secondo esperimento è il confronto di performance tra simple frequency method (SF), collaborative filtering(CF), single level AR e multi level AR. Il SF è un metodo basico che suggerisce I top-N elementi più popolari tra tutti gli utenti contando la frequenza di preferenza per ogni elemento nella matrice di preferenza.

**In CF, the cosine similarity was used as a similarity measure, and the number of similar users was set to 20 (which was the best among 10, 20, 30, 40, and 50). In AR+, we used all available association rules because applying higher *minSup* and *minCon* to select strong rules make the number of applicable association rules decreased rapidly, and causes poor performance due to sparseness of dataset. In MAR, we used up to level 1 category information for MovieLens dataset and up to level 2 category for KDD dataset. MAR showed better performance as *minSup* and *minCon* thresholds for high-level categories are set to higher values. We set both thresholds to 80% for two datasets respectively in this experiment.**

In CF è stata usata cosine come misura di similarità. Il numero di utenti simili è stato scelto a 20. In AR sono state usate tutte le regole di associazione perchè applicando il più alto minSup e minCon per selezionare le forti regole, il loro numero calava drasticamente con scarse performance. In MAR sono state usate le informazioni del primo livello di categoria per movieLens e il livello 2 per KDD. MAR ha mostrato migliori performance come livelli di minSup e minCon per I livelli superiori di categoria. Risultati superiori all'80% per ogni esperimento.

**In all of experiments, we used N = 10, 20, 30, 40, and 50, as the number of items to be recommended and measured Recall for each N respectively. All experiments were performed on Pentium-4 running at 1.8GHz, 1Gbytes of memory, and Windows 2000 operating system.**

In tutti gli esperimenti sono stati usati N = 10, 20, 30, 40, and 50 come numero di elementi da raccomandare e misurata Recall per ogni N.

**3.3. Results**

**3.3.1. Applying methods of association rules. We compared the two different methods of applying association rules. AR-MAX is the method of applying the maximum confidence rule, and AR+ is the method of applying the sum of confidences of all available rules. The results show that AR+ outperforms AR-MAX by 20% for MovieLens dataset and by 7% for KDD dataset in all cases of top-N. This result means that the item associated with more rules is more appropriate to recommend.**

Sono stati paragonati due differenti metodi di applicazione di AR. AR-MAX con la massima conficenza, AR+ con la somma delle confidenze delle regole disponibili. I risultati mostrano che AR+ supera del 20% AR-MAX per movieLens ed è superato del 7% da KDD. Questo dimostra che gli elementi associati con pi+ regole sono migliori per la raccomandazione.

**3.3.2. Performance of various algorithms.**

**Figure 1 and 2 show the results of four different recommendation algorithms for two datasets.**

**For the MovieLens dataset, when we computed the recommendation vector in MAR, we used 0 = 0.9 and 1 = 0.1 as the weight for each level. MAR showed relatively good performance for small values of 1. In Figure 1, the result shows that MAR achieved 2.5%, 1.2%, 1.3%, 1.8%, and 2.0% higher accuracy for each value of top-N compared with AR+. This result shows that the application of multi-level association rules is effective. MAR showed somewhat less performance compared with CF when N is less than 20, but showed higher accuracy by 1.5%, 5.4%, and 8.6% when N is more than 30. Performance of CF becomes relatively poor as N increases because the CF method can’t predict preference of many items due to data sparseness.**

Figure mostra i risultati dei differenti algoritmi per i differenti dataset. Per ML, quando è stato calcolato MAR si è usato alfa0 = 0.9 e alfa1 = 0.1 come peso di ogni livello. MAR mostra dei buoni risultati soprattuttoi per valori piccoli di alfa1. Nella figura 1si nota che MAR funziona meglio di tutti gli altri. Accusa un po' rispetto a CF quando N è minore di 20 ma migliore sopra a N=30. Il problema di CF è che con tanti item predice male.

**Figure 2 shows the result of the KDD dataset. We used 0 = 0.6, 1 = 0.3, and 2 = 0.1 as the weight for each level. MAR outperformed AR+ by average 90% for all top-N and CF up to 43% when N is more than 30. In the CF method, performance was almost the same when N is more than 30. It also revealed the sparseness problem of CF. Especially, AR+ shows less performance than SF for all top-N. The KDD data set was so sparse that the number of association rules was very small.**

La figura 2 mostra il risultato su KDD dataset. Il dataset è molto sparso e quindi il numero di regole di associazione è molto piccolo.

**3.3.3. Recommendation time.**

*The computation times for recommendation for 10000 users were*

*0.501 sec for CF, and*

*0.011 sec for MAR in case of EachMovie dataset,*

*and*

*0.072 sec for CF*

*and 0.00037 sec for MAR in case of KDD dataset.*

*The experimental results with 20000 and 30000 users also show that the model-based method MAR took almost constant time, but time of memory-based method CF increases as the number of users grows for two dataset respectively*

**Conclusioni:**

In this paper, we present a model-based recommendation algorithm that uses multi-level association rules to alleviate the sparseness and scalability problems in memory-based recommendation. In this algorithm, we build a model for preference prediction by using association rule mining. Multi-level association rules are used to compute preferences for items that are not covered by the association rules between items due to the data sparseness.

Through the experiments using MovieLens and KDD dataset, it is shown that applying multi-level association rules increases recommendation accuracy compared with applying single-level association rules only. The suggested algorithm also shows better performance compared with the basic collaborative filtering as the number of recommendation increases in a sparse environment.