Retrieval in CBR Using a Combination of Similarity and Association Knowledge

Yong-Bin Kang[1], Shonali Krishnaswamy[1,2], and Arkady Zaslavsky[1,3]

[1] Faculty of IT, Monash University, Australia
{yongbin.kang,shonali.krishnaswamy}@monash.edu
[2] Institute for Infocomm Research (I^{2}R), Singapore
[3] Information Engineering Laboratory, ICT Centre, CSIRO, Australia
arkady.zaslavsky@csiro.au

Abstract. Retrieval is often considered the most important phase in Case-Based Reasoning (CBR), since it lays the foundation for the overall performance of CBR systems. In CBR, a typical retrieval strategy is realized through similarity knowledge and is called similarity-based retrieval (SBR). In this paper, we propose and validate that association analysis techniques can be used to enhance SBR. We propose a new retrieval strategy USIMSCAR that achieves the retrieval process in CBR by integrating similarity and association knowledge. We evaluate USIMSCAR, in comparison with SBR, using the Yahoo! Webscope Movie dataset. Through our evaluation, we show that USIMSCAR is an effective retrieval strategy for CBR that strengthens SBR.

1 Introduction

The premise of CBR is that experience in the form of past cases can be leveraged to solve new problems. In CBR, experiences are stored in a database known as a *case base*, and an individual experience is called a *case*. Typically, there are four well-organized phases adopted in CBR [1]: *Retrieve* one or several cases considered useful for solving a given target problem, *Reuse* the solution information of the retrieved cases, *Revise* the solution information to better fit the target problem, and *Retain* the new solution once it has been confirmed or validated.

Retrieval is considered a key phase in CBR, since it lays the foundation for overall performance of CBR systems [2]. Its aim is to retrieve *useful* cases that can be successfully used to solve a new problem. If the retrieved cases are not useful, CBR systems will not eventually produce any good solution for the new problem. To achieve the retrieval process, CBR systems typically rely on a retrieval strategy that exploits *similarity knowledge* and is referred to as *similarity-based retrieval* (SBR) [3]. In SBR, similarity knowledge aims to approximate the *usefulness* of stored cases with respect to the target problem [4]. This knowledge is usually encoded in the form of similarity measures used to compute similarities between a new problem and the cases. By using similarity measures, SBR finds cases with higher similarities to the new problem, and then their solutions are utilized to solve the problem. Thus, it is evident that SBR tends to rely strongly

J. Tang et al. (Eds.): ADMA 2011, Part I, LNAI 7120, pp. 1–14, 2011.

on similarity knowledge, ignoring other forms of knowledge that can be further leveraged for improving the retrieval performance [3,4,5,6].

In this paper, we propose that association analysis of stored cases can improve traditional SBR. We propose a new retrieval strategy USIMSCAR that leverages *association knowledge* in conjunction with similarity knowledge. Association knowledge is aimed to represent certain interesting relationships, shared by a large number of cases, acquired from stored cases using association rule mining. We show USIMSCAR improves SBR through an experimental evaluation using the "Yahoo! Webscope Movie" dataset. This paper is organized as follows. Section 2 presents our research motivation. Section 3 reviews the related work. Section 3 presents a background of similarity knowledge and association knowledge. Section 4 presents our approach for extracting and representing association knowledge. Section 5 presents the USIMSCAR algorithm. Section 6 evaluates USIMSCAR in comparison with SBR. Section 7 presents our conclusion and future research directions.

2 Motivation

To illustrate our research motivation, we use a medical diagnosis scenario presented in [7]. Consider a case base $\mathcal{D}$ that consists of five patient cases P_1, ..., P_5 shown in Table 1. Each case is represented by a problem described by 5 attributes (symptoms) $A_1, ..., A_5$, and a corresponding solution described by an attribute (diagnosis) A_6. Our aim is to determine the correct diagnosis for a new patient Q. We note that Q was suffering from 'appendicitis' as specified in [7], and this therefore represents the correct diagnosis.

Table 1. A patient case base

Cases	Local Pain(A_1)	Other Pain(A_2)	Fever (A_3)	Appetite Loss(A_4)	Age (A_5)	Diagnosis (A_6)	Similarity to Q
p_1	right flank	vomit	38.6	yes	10	appendicitis	0.631
p_2	right flank	vomit	38.7	yes	11	appendicitis	0.623
p_3	right flank	vomit	38.8	yes	13	appendicitis	0.618
p_4	right flank	sickness	37.5	yes	35	gastritis	**0.637**
p_5	epigastrium	nausea	36.8	no	20	stitch	0.420
Q	right flank	nausea	37.8	yes	14	?	
Weight	0.91	0.78	0.60	0.40	0.20		

To predict a diagnosis for Q, SBR retrieves the most similar cases to Q by identifying the cases whose attributes are similar to those of Q using a similarity metric. We use the following metric, the same one used in the work [7], measuring the similarity between Q and each case $p \in \mathcal{D}$,

$$SIM(Q,p) = \frac{\sum_{i=1}^{n} w_i \cdot sim(q_i, p_i)}{\sum_{i=1}^{n} w_i},$$

$$sim(q_i, p_i) = \begin{cases} 1 - \frac{|q_i - p_i|}{A_i^{\max} - A_i^{\min}}, \text{ if } A_i \text{ is numeric,} \\ 1, \text{ if } A_i \text{ is discrete } \& \ q_i = p_i, \\ 0, \text{ otherwise,} \end{cases} \tag{1}$$

where w_i is a weight assigned to an attribute A_i, q_i and p_i are values of A_i of Q and p respectively, n is the number of attributes of Q and p (i.e. n=5), $sim(q_i, p_i)$ denotes a similarity measure between q_i and p_i, and $A_i^{\max}$ and $A_i^{\min}$ are the maximum and minimum values, respectively, that A_i takes on. Using the above metric, assume that SBR chooses the most similar case to Q. As seen in Table 1, p_4 is thus chosen, since it is the most similar case to Q. It means that a diagnosis choice for Q is 'gastritis'. But it turned out to be wrong, since Q suffered from 'appendicitis' as mentioned above. To overcome the problem, our idea is to extract, represent, and exploit the knowledge of how known problems are highly associated with known solutions in $\mathcal{D}$. In $\mathcal{D}$, we may obtain the knowledge that the problems of cases p_1, p_2 and p_3 are highly associated with 'appendicitis', while those of a case p_4 with 'gastritis'. The former association strength S_1 may be higher than the latter one S_2, since S_1 is supported by three cases, while S_2 by a single case. If such strength were to be appropriately quantified, and combined with the similarities in shown Table 1, a diagnosis for Q can be more accurately determined. This is the key idea of our proposed USIMSCAR.

3 Related Work

SBR has been widely used in various CBR application domains, such as medical diagnosis [8] and product recommendation [9], to predict useful cases with respect to the target problem Q. It is typically implemented through *k-nearest neighbor retrieval* or simply k-NN [2]. In a CBR context, the idea of k-NN is that the retrieval process in CBR is achieved through retrieving the k most similar cases to Q. Thus, the quality of the employed similarity measures for determining those cases is an important aspect in k-NN. Over the years, researchers have studied k-NN to enhance its accuracy. For example, it is shown that k-NN can be integrated with feature selection (FS) [10]. FS is a technique for determining relevant features (or attributes) from the original features of cases. k-NN is easily extended to include FS by only considering relevant features when computing the similarity between Q and each case.

To enhance SBR, much work has also focused on integrating SBR with *domain knowledge* and *adaptation knowledge*. For example, Stahl [4] proposes a retrieval approach in which similarity assessment during SBR is integrated with domain knowledge. Aamodt [11] proposes an approach that cases are enriched with domain knowledge that guides the retrieval of relevant cases. Adaptation knowledge is also used to enhance SBR in which this knowledge indicates whether a case can be easily modified to fit the new problem [3]. In this approach, matches between the target problem and cases are done, only if there is evidence in adaptation knowledge that such matches can be catered for during retrieval.

Our approach for enhancing SBR differs from the above approaches in three aspects: (1) While many kinds of learnt and induced knowledge has been utilized, we leverage association knowledge that has not been used for retrieval in CBR systems. (2) The acquisition of both domain and adaptation knowledge is usually known as a very complex and difficult task, thus often leads to knowledge bottleneck phenomenon [4]. However, association knowledge acquisition is straightforward, since it is automatically acquired from stored cases, a fundamental knowledge source in CBR, using association rule mining. (3) Association knowledge extraction is achieved through capturing strongly evident associations between known problem features and solutions shared by a large number of cases. This scheme can be compared to FS, since in a CBR context it mainly focuses on estimating the relevance of problem features highly correlated to known solutions. However, FS usually assumes feature independence, ignoring identifying interesting relationships between problem features, dependent on each other, and each solution. In contrast, association knowledge extraction includes and considers all interesting frequent patterns and association structures from a given case base using association rule mining.

4 Background of Similarity and Association Knowledge

Prior to presenting our proposed USIMSCAR, we provide a background of similarity and association knowledge. We first present our case representation scheme that is the basis for representing both similarity and association knowledge. To represent cases, many CBR systems generally adopt well-known knowledge representation formalisms, such as *attribute-value pairs* and *structural* representations [4]. In our work, we choose the attribute-value pairs representation due to its simplicity, flexibility and popularity. Let $A_1, ..., A_m$ be attributes defined in a given domain. An *attribute-value pair* is a pair (A_i, a_i), where A_i is an attribute (or feature[1]) and a_i is a value of $A_{i \in [1,m]}$. A *case* C is a pair $C = (X, Y)$ where X is a problem, represented as $X = \{(A_1, a_1), ..., (A_{m-1}, a_{m-1})\}$, and Y is the solution of X, represented as $Y = (A_m, a_m)$. We call an attribute A_m a *solution-attribute*. A *case base* $\mathcal{D}$ is a collection of cases.

4.1 Background of Similarity Knowledge

In a CBR context, we refer to similarity knowledge as knowledge encoded via measures computing the similarities between the target problem and stored cases. To formulate the measures, CBR systems often use a widely used principle. This is the *local-global principle* that decomposes a similarity measure by *local similarities* for individual attributes, and a *global similarity* aggregating the local similarities [4]. An accurate definition of local similarities relies on attribute types. A global similarity function can be arbitrarily complex, but usually simple functions (e.g. weighted average) are used in many CBR systems. Referring to Eq. 1 SIM is a global similarity function, and sim is a local similarity function.

4.2 Background of Association Knowledge

Our premise is that SBR can be enhanced by the inclusion of association knowledge representing evidently interesting relationships shared by a large number of stored cases. It is extracted from stored cases and represented using *association*

[1] To simplify the presentation, we do not distinguish between terms "attributes" and "features", and use these terms interchangeably.

rule mining [12], *class association rule mining* [13] and *soft-matching criterion* [14], which are outlined in the following.

Association rule mining [12] aims to mine certain interesting associations in a transaction database. Let I be a set of distinct literals called *items.* A set of items $X \subseteq I$ is called an itemset. Let $\mathcal{D}$ be a set of transactions. Each transaction $T \in \mathcal{D}$ is a set of items such that $T \subseteq I$. We say that T *contains* an itemset X, if $X \subseteq T$ holds. Every *association rule* has two parts: an *antecedent* and a *consequent.* An association rule is an implication of the form $X \rightarrow Y$, where $X \in I$ is an itemset in the antecedent and $Y \in I$ is an itemset in the consequent, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ has *support* s in $\mathcal{D}$ if $s\%$ of transactions in $\mathcal{D}$ contain $X \cup Y$. This holds in $\mathcal{D}$ with *confidence* c if $c\%$ of transactions in $\mathcal{D}$ that contain X also contain Y. Association rule mining can also be used for discovering interesting relationships among stored cases. In a CBR context, a transaction can be seen as a case, and an item as an attribute-value pair. Referring to Table 1, we can mine a rule $r_1 : (A_1, \text{right flank}) \rightarrow (A_2, \text{vomit})$. Let X be an item (A_1, right flank). Let Y be an item (A_2, vomit). The support of r_1 is 0.6, since X and Y occur together in three out of five cases in $\mathcal{D}$. The confidence of r_1 is 0.75, since Y occurs in three out of four cases that contain X in $\mathcal{D}$. Apriori [12] is one of the traditional algorithms for association rule mining.

Class association rules (cars) [13] is a special subset of association rules whose consequents are restricted to a single target variable. In a CBR context, cars can be seen as special association rules whose consequents only hold special items formed as pairs of a "solution-attribute" and its values. We call such an item a *solution-item.* Thus, a car has the form $X \rightarrow y$, where $X \subseteq I$ an itemset in the antecedent and $y \in I$ is a solution-item in the consequent. Our aim of building association knowledge is to represent the knowledge encoding how certain known problems are associated with known solutions in a case base. For the purpose, we use the form of cars, since it is suited for this goal. Note that the car $X \rightarrow y$ encodes an association between an itemset X (i.e. attribute-value pairs of known problems), and a solution-item y (i.e. the corresponding solution).

Consider the association rule $X \rightarrow Y$. A limitation of traditional association rule mining algorithms (e.g. Apriori [12]) is that itemsets X and Y are discovered using equality relation. Unfortunately, when dealing with items similar to each other, these algorithms may perform poorly. For example, a supermarket sales database, Apriori cannot find rules like "80% of the customers who buy products similar to milk (e.g. cheese) and products similar to eggs (e.g, mayonnaise) also buy bread." To address this issue, SoftApriori [14] was proposed. It uses the *soft-matching criterion*, where the antecedent and the consequent of association rules are found using similarity assessment. By doing so, this criterion can be used to model richer relationships among case features than the equality relation.

5 Association Knowledge Formalization

This section presents our approach for extracting and representing association knowledge used the techniques outlined in Section 3. The aim of building association knowledge is two-fold. The first is to represent strongly evident, interesting associations between known problem features and solutions shared by a large number of cases. The second is to leverage these associations along with similarity knowledge in our proposed USIMSCAR to improve SBR.

We propose to represent association knowledge via cars whose antecedents are determined by applying the soft-matching criterion. We refer to these rules as *soft-matching class association rules* (scars). A scar $X \rightarrow y$ implies that the target problem Q is likely to be associated with the solution contained in an item y, if the problem features of Q are highly similar to an itemset X.

Let $\mathcal{D}$ be a set of cases, where each case is characterized by attributes $A_1, ..., A_m$. We call a pair $(A_i, a_i)_{i \in [1,m-1]}$ an *item.* We call a pair (A_m, a_m) a *solution-item.* Let I be a set of items. A set $L \subseteq I$ with $k = |L|$ is called a k-itemset or simply an itemset. Let $sim(x, y)$ be a function computing the similarity between two items $x, y \in I$ in terms of their values. We say that x and y are similar, iff $sim(x, y) \geq$ *a user-specified minimum similarity* (minsim). Given two itemsets $X, Y \subseteq I$ ($|X| \leq |Y|$), $ASIM(X, Y)$ is a function that computes the asymmetric similarity of X with respect to Y, defined as $\sum_{x \in X, y \in Y} \frac{sim(x,y)}{|X|}$, where x, y are items with the *same* attribute label. Let X_1 be a 2-itemset $\{(A_1, a), (A_2, b)\}$. Let Y_1 be a 2-itemset $\{(A_1, a'), (A_2, b')\}$. Assuming similarity functions for A_1 and A_2 are denoted as sim_{A_1} and sim_{A_2} respectively, $ASIM(X_1, Y_1) = (sim_{A_1}(a, a') + sim_{A_2}(b, b'))/2$. We say that X is a soft-subset of Y ($X \subseteq_{soft} Y$), iff $ASIM(X, Y) \geq$ minsim; or Y *softly contains* X. The *soft-support-sum* of an itemset $X \subseteq I$ is defined as the sum of the asymmetric similarities of X with respect to cases in $\mathcal{D}$ that softly contain X, $softSuppSum(X) = \sum_{X \subseteq_{soft} C \in \mathcal{D}} ASIM(X, C)$. The *soft-support* of X is defined as $softSupp(X) = softSuppSum(X)/|\mathcal{D}|$. The *soft-support-sum* of a rule $X \rightarrow y$ is defined as the sum of the asymmetric similarities of X with respect to cases in $\mathcal{D}$ that softly contain X and contain y, $softSuppSum(X \rightarrow y)$. The *soft-support* of this rule is defined as $softSupp(X \rightarrow y) = softSuppSum(X \rightarrow y)/|\mathcal{D}|$. The *soft-confidence-sum* of a rule $X \rightarrow y$ is defined as the sum of the asymmetric similarities of X with respect to cases in $\mathcal{D}$ that softly contain X also contain y, $softConfSum(X \rightarrow y)$. The *soft-confidence* of this rule is defined as $softConf(X \rightarrow y) = softConfSum(X \rightarrow y)/|\mathcal{D}|$.

The key operation for scars mining is to find all ruleitems that have soft-supports $\geq$ (*a user-specified minimum support*) (minsupp). We call such ruleitems *frequent* ruleitems. For all the ruleitems that have the same itemset in the antecedent, one with the highest *interestingness* is chosen as a *possible rule* (PR). To measure the interestingness of association rules, support and confidence are typically used. On some occasions, a combination of them is used. Often, a rationale for doing so is to define a single optimal interestingness by leveraging their correlations. We choose the Laplace measure (LM) [15] that combines soft-support and soft-confidence such that they are monotonically related. Given a ruleitem $r : X \rightarrow y$, its LM $Laplace(r)$ can be denoted as $\frac{|\mathcal{D}| \cdot softSupp(X \rightarrow y) + 1}{|\mathcal{D}| \cdot softSupp(X \rightarrow y)/softConf(X \rightarrow y) + 2}$. If $Laplace(r) \geq$ *a user-specified minimum level of interesting* (min-interesting), we say r is *accurate.* A candidate set of scars consists of all the PRs that are frequent and accurate.

Let k-ruleitem be a ruleitem whose antecedent has k items. Let F_k be a set of frequent k-ruleitems. The following is a description for the scars mining algorithm: (1) For 1-ruleitems $X \subseteq I$, we find $F_1 = \{\{X\} | softSupp(X) \geq$ minsupp$\}$. A set $SCAR_1$ is then generated by only choosing PRs from F_1. (2) For each k subsequent pass, we find a set of new possibly frequent ruleitems CR_k using F_{k-1} found in the $(k-1)^{th}$ pass. We then generate a new set F_k by extracting ruleitems in CR_k whose soft-support $\geq$ minsupp. A set $SCAR_k$ is generated by only choosing PRs from F_k. (3) From $SCAR_1$, ..., $SCAR_k$, we choose only sets whose $i \in [1, k] \geq$ *a user-specified minimum ruleitem size* (minitemsize), and store them in a set $SCARS$. Our idea is to choose a small representative subset of frequent ruleitems from the large number of resulting frequent ruleitems. The longer the frequent ruleitem, the more significant it is [16]. We perform a rule pruning on ruleitems in $SCARS$. A rule r is pruned, if $Laplace(r) <$ min-interesting. The set of ruleitems after the pruning is finally returned as the set of scars to be used in our proposed USIMSCAR.

6 The USIMSCAR Algorithm

This section presents USIMSCAR that leverages both association and similarity knowledge to enhance SBR. The main challenge is how to combine similarity and association knowledge appropriately and effectively, thereby strengthening the retrieval performance of SBR. This section address this challenge by presenting the USIMSCAR algorithm. The rationale for leveraging association knowledge in USIMSCAR falls into two objectives: (1) enhancing the usefulness of the cases, retrieved by using similarity knowledge as with SBR, with respect to a new problem Q by including both similarity and association knowledge, and (2) directly leveraging a number of scars whose usefulness is relatively high with respect to Q, eventually utilizing such scars with their usefulness in USIMSCAR.

Given a new problem Q, USIMSCAR's goal is to produce a retrieval result RR consisting of objects that can be used to solve Q by leveraging similarity and association knowledge. Such objects are obtained from both stored cases and scars mined. Let $\mathcal{D}$ be a set of cases. Let $SCARS$ be the set of scars mined from $\mathcal{D}$. Below we present the USIMSCAR algorithm.

(1) From $\mathcal{D}$, we find the k most similar cases to Q, and store them in a set RC. We denote $SIM(Q, C)$ as the similarity between Q and a case C.

(2) In $SCARS$, we find the k' most similar scars to Q, and store them in a set RS. A question raised here is how to compute the similarity $SIM(Q, r)$ between Q and a scar r. Its answer lies in our choice of cars representation for scars mining. Note that scars have the *identical* structure as cases: the antecedent and consequent correspond to the problem and solution part of cases respectively. Thus, $SIM(r, Q)$ can be defined in the same way as $SIM(Q, C)$ in (1). To generate RS, we only consider scars (RCS) in $SCARS$ such that their antecedents are soft-subsets of cases in RC, rather than all scars in $SCARS$ for efficiency. Each case $C \in RC$ is chosen as a similar case to Q ($C \sim Q$). Assuming each scar $r \in RCS$ has the form $r : X \rightarrow y$, X is a soft-subset of C ($X \subseteq_{soft} C$). Since

$C \sim Q$ and $X \subseteq_{soft} C$, $X \subseteq_{soft} C \sim Q$ can be derived. It implies that RCS is a particular subset (i.e. soft-subset) of cases in RC similar to Q.

(3) For each case $C \in RC$, we select a scar $r_C \in SCARS$. It is chosen if it has the highest interestingness among those scars in $SCARS$ such that their antecedents are soft-subsets of C and their consequents are equal to the solution of C. We then quantify the usefulness of C with respect to Q ($USF(C, Q)$) by $SIM(C, Q) \times \mathit{Laplace}(r_C)$. If candidates for r_C are chosen more than one, say m, we use the average of the interestingness of these m scars to compute $\mathit{Laplace}(r_C)$. If there is no candidate for r_C, we use min-interesting for $\mathit{Laplace}(r_C)$. Note that in SBR, the usefulness of C regarding Q is measured by $SIM(C, Q)$. Our combination schemes aims to quantify this usefulness by leveraging $SIM(C, Q)$ and $\mathit{Laplace}(r_C)$. We then cast C to a *generic object* O that can hold any cases and scars. O has two fields: $O.inst = C$, $O.usf = USF(C, Q)$. The object O is then added to a retrieval result RR.

(4) For each scar $r \in RS$, we quantify the usefulness of r with respect to Q ($USF(r, Q)$) by $SIM(r, Q) \times \mathit{Laplace}(r_C)$. This aims to quantify the usefulness by combining $SIM(r, Q)$ obtained from similarity knowledge and $\mathit{Laplace}(r_C)$ acquired from association knowledge. We then cast r to a generic object O with two fields: $O.inst = r$, $O.usf = USF(r, Q)$. The object O is then added to RR.

(5) We further enhance the usefulness of each object $O \in RR$ using the frequency of *solution occurrence* among objects in RR. Our premise is that if O's solution is more frequent in RR, O is more useful in RR. If O is cast from a case C, its solution means C's solution. If O cast from a scar r, its solution is r's consequent. Let S be a set of solutions of objects in RR. Let S_O be a set of objects in RR that have the solution equal to the solution of an object $O \in RR$. For each object $O \in RR$, we compute $\delta(S_O)$ as $\delta(S_O) = |S_O|/|RR|$ Finally, we enhance $O.usf$ by multiplying $\delta(S_O)$. Eventually, each object $O \in RR$ with $O.usf$ is utilized to induce a solution for Q.

We now illustrate how USIMSCAR operates using the case base $\mathcal{D}$ shown in Table 1. From $\mathcal{D}$, we can generate 4 scars shown in Table 2 using the similarity SIM in Eq. 1.

Table 2. The scars generated

Rules	*Laplace*	Soft-subset of
r_1: $\{(A_1$,right flank),$(A_2$,vomit),$(A_3$,38.6),$(A_4$,yes),$(A_5$,13)$\}$ → $(A_6$,appendicitis)	0.922	p_1, p_2, p_3
r_2: $\{(A_1$,right flank),$(A_2$,vomit),$(A_3$,38.7),$(A_4$,yes),$(A_5$,10)$\}$ → $(A_6$,appendicitis)	0.922	p_1, p_2, p_3
r_3: $\{(A_1$,right flank),$(A_2$,vomit),$(A_3$,38.8),$(A_4$,yes),$(A_5$,10)$\}$ → $(A_6$,appendicitis)	0.922	p_1, p_2, p_3
r_4: $\{(A_1$,right flank),$(A_2$,sickness),$(A_3$,37.5),$(A_4$,yes),$(A_5$,35)$\}$ → $(A_6$,gastritis)	0.775	p_4

Using the above scars, USIMSCAR takes the following steps (assume k=k'=2): (1) It finds the 2 most similar cases to Q: $RC = \{p_4, p_1\}$ for $SIM(Q, p_4)$=0.637, $SIM(Q, p_1)$=0.631. (2) It finds the 2 most similar scars to Q: $RS = \{r_1, r_4\}$ for $SIM(Q, r_1)$=0.640, $SIM(Q, r_4)$=0.637. (3) For each case $C \in RC$, r_C is chosen. For p_4, r_4 is selected. For p_1, r_1, r_2 and r_3 are selected. Then, $USF(Q, p_4)$ and $USF(Q, p_1)$ are quantified as $USF(Q, p_4)$=0.494, $USF(Q, p_1)$=0.581. Then, p_4 with $USF(Q, p_4)$ and p_1 with $USF(Q, p_1)$ are cast to new objects and stored in a set RR. (4) For each scar $r \in RS$, its usefulness to Q is quantified as $USF(Q, r_1)$=0.594, $USF(Q, r_4)$=0.496. Then, these scars with their usefulness are cast to new objects and stored in RR. (5) Assume that each object in RR has another field s holding its solution. RR has 4 objects $RR = \{O_1, ..., O_4\}$ shown in Table 3. As observed, there are only two sets of objects regarding solutions. For each object $O \in RR$, $O.usf$ is enhanced by weighting $\delta(S_O) = |S_O|/|RR|$. The enhancement results are shown under the column 'final usf' in the table. Eventually, if we choose the most useful one to Q, we retrieve O_3 and its solution 'appendicitis', Q really had, is used as a diagnosis for Q.

Table 3. The retrieval result RR

field: inst	field: usf	field: solution	final usf
$O_1.inst = p_4$,	$O_1.usf = 0.494$,	$O_1.s =$ gastritis	0.247
$O_2.inst = p_1$,	$O_2.usf = 0.581$,	$O_2.s =$ appendicitis	0.291
$O_3.inst = r_1$,	$O_3.usf = 0.594$,	$O_3.s =$ appendicitis	**0.297**
$O_4.inst = r_4$,	$O_4.usf = 0.496$,	$O_4.s =$ gastritis	0.248

7 Evaluation

We experimentally show that USIMSCAR improves SBR with respect to retrieval performance. Our work has focused on proposing a new retrieval strategy for CBR. Thus, as a target application task, it is desirable to choose a task that is highly dependent on retrieval performance in a CBR context. One suitable task is case-based classification [17], defined as: given a new problem Q, its goal is to find similar cases to Q from a case base, and classify Q based on the retrieved cases. Thus, in principle, this approach is strongly dependent on the result obtained through retrieval in CBR.

As target SBR approaches to be compared with USIMSCAR, we choose the following k-NN approaches implemented in Weka, since SBR is typically implemented through k-NN: (1) IB1 is the simplest form of k-NN using the Euclidean distance to find the most similar case C to Q. (2) IBkBN extends IB1 by using the best k (i.e. the number of the most similar cases) determined by cross-validation. (3) IBkFS extends IBkBN by using a feature selector CfsSubsetEval available in Weka. (4) KStar is an implementation of K* [18], where similarity for finding the most similar cases to Q is defined through entropy.

In a k-NN approaches context, classification has two stages. The first is to find similar cases RR to Q using similarity knowledge, and the second is to classify Q using the solutions in RR. In a USIMSCAR context, the first is to find a set of "useful cases and rules" RR using "similarity and association knowledge", and the second is to classify Q using the solutions of objects in RR. Our work is focused on the first stage. The second stage can be achieved using *voting*. Due to generality, we adopt *weighted voting*, where objects in RR get to vote, on the solution of Q, with votes weighted by their significance to Q. For each object in RR, in SBR the significance is measured using its similarity to Q, while in USIMSCAR it is measured using its usefulness with respect to Q.

We use the Yahoo! Webscope Movie dataset (R4) usually used for evaluating recommender systems. In a CBR context, each instance in R4 has the form (x, s_x): x is a problem description characterized by two user attributes (birthyear, gender) and ten movie attributes (see Table 4), and s_x is the corresponding solution meaning a rating assigned to a movie by a user. Before testing, we removed the instances that contain any missing values of any movie attributes, and redundant movie attributes (e.g. actors are represented using both 'actor id' and 'name', so we included only the name). Finally, R4 consists of training data (74,407 ratings scaled from 1 to 5 rated by 5,826 users for 312 movies), and testing data (2,705 ratings scaled from 1 to 5 rated by 993 users for 262 movies).

Table 4. Movie information (movie-info)

Attributes	Description	Type
title	movie title	String
synopsis	movie synopsis	String
mpaa_rating	MPAA rating of movie	Nominal
genres	list of the genres of movie	Set-valued
directors	list of the directors of movie	Set-valued
actors	list of the actors of movie	Set-valued
avg-critic-rating	average of the critic reviews of movie	Numeric
rating-from-Mom	rating to movies obtained from the Movie Mom	Numeric
gnpp	Global Non-Personalized Popularity (GNPP), of movie, computed by Yahoo! Research	Numeric
avg-rating	average movie rating by users in the training data	Numeric

For each instance Q in the testing data, our goal is to predict the correct rating that the user will be likely to rate using the training data. We split it into three classification tasks taking a user and a movie, and classify a rating in three rating-scales: RS(5) is a five rating-scale [1,5], RS(3) is a 3 rating-scale where a rating indicates whether a movie would be *liked* (> 3), *normal* (=3) or *disliked* (<3), and RS(2) is a 2 rating-scale where a rating indicates whether a movie would be *liked* (>3) or *disliked* ($\leq$3). We evaluate the prediction using *classification accuracy* (CA) and *predictive accuracy* (PA) that are widely used for classification and recommendations. CA measures the proportion of correctly classified instances over all the instances tested. PA measures how close predicted ratings are to the actual user ratings. *Mean absolute error* (MAE) is widely used to measure this accuracy, $\sum_{i=1}^{N} |p_i - r_i|/N$, where p_i is a predicted rating, r_i is an actual rating for an instance i, and N is the number of instances tested. Regarding MAE, lower values are more accurate. We compute the MAE values for each user in the testing data, and then average over all users in the data.

The similarity knowledge used is encoded as a similarity measure using the global-local principle. Given a new problem Q and a case, their global similarity is defined as SIM in Eq. 1, and local similarities are defined on four types. For numeric and nominal attributes, we used sim in Eq. 1. For set-valued attributes, we used the *Jaccard coefficient*. For string attributes, we converted a given string into a set-valued representation by tokenizing it, and applied the Jaccard coefficient. We implemented IBkBN, IBkFS and USIMSCAR to be working with SIM

to find the k most similar cases for Q. The function SIM is also used to find the k' most similar scars with respect to Q in USIMSCAR. For the approaches, we chose a best value for k using cross-validation from 1 to 15. We observed that increasing k beyond 15 hardly changed the results.

To generate scars from R4, we set minsim, min-interesting, and minitemsize to arbitrary values 0.95 (95%), 0.7, and 7 respectively. Setting a value for minsupp is more complex, since it has a stronger effect on the quality of USIMSCAR. If minsupp is set too high, those possible scars, which cannot satisfy minsupp but with high interestingness (Laplace measure) values, will not be included. While if minsupp is set too low, it is possible to generate too many scars including trivial rules. Both occasions may lead to a reduction in the performance of USIMSCAR. From our experiments, we observed that once minsupp is set to 0.1, the performance of USIMSCAR is best. We thus set a value for minsupp to 0.1.

7.1 Results and Analysis

We now present the experimental results of USIMSCAR and the compared k-NN approaches (simply 4KNN) in terms of both classification accuracy (CA) and MAE in Tables 5 and 6. For each rating-scale, the best accuracy is denoted in boldface. The mark "$\star$" indicates that USIMSCAR attains a significant improvement over the target measure. For CA, it is discovered by the Z-test [19] with 95% confidence, and for MAE by the paired t-tests [19] at 95% confidence. Each number in parentheses denotes the improvement ratio of USIMSCAR over the target measure.

Table 5 indicates that USIMSCAR achieves 100% better performance than 4KNN in all rating-scales in terms of CA. We find that the 91.6% comparisons between USIMSCAR and 4KNN are statistically significant in terms of CA.

Table 5. The classification accuracy results

Compared Classifiers	Classification Accuracy(%)		
	RS(5)	RS(3)	RS(2)
IB1	46.30 (2.76% $\star$)	72.95 (6.61% $\star$)	75.24 (5.02% $\star$)
IBkBN	48.61 (0.45%)	75.97 (3.59% $\star$)	77.12 (3.14% $\star$)
IBkFS	46.28 (2.78% $\star$)	74.17 (5.39% $\star$)	75.24 (5.02% $\star$)
KStar	44.34 (4.72% $\star$)	74.37 (5.19% $\star$)	75.07 (5.19% $\star$)
USIMSCAR	**49.06**	**79.56**	**80.26**

Table 6. The MAE results

Compared Classifiers	Predictive Accuracy (MAE)		
	RS(5)	RS(3)	RS(2)
IB1	.9139 (24.14% $\star$)	.3760 (8.76% $\star$)	.2476 (5.02% $\star$)
IBkBN	.8532 (18.07% $\star$)	.3482 (5.98% $\star$)	.2288 (3.08% $\star$)
IBkFS	.8392 (16.67% $\star$)	.3541 (6.57% $\star$)	.2376 (4.02% $\star$)
KStar	.8710 (19.89% $\star$)	.3652 (4.02% $\star$)	.2493 (5.19% $\star$)
USIMSCAR	**.6725**	**.2884**	**.1974**

As shown in Table 6, we also find that USIMSCAR achieves 100% better performance than 4KNN in all rating-scales in terms of MAE. All the improvements are deemed to be statistically significant. Through these results, we demonstrate that USIMSCAR has the ability to retrieve more useful objects (i.e. cases and scars) with respect to the target problems than SBR. As outlined in the USIMSCAR algorithm, these objects are identified and quantified by using a combination of similarity and association knowledge. This further establishes the validity of the primary motivation of this research that the combination will lead to improving SBR. The real strength of our evaluation lies in the fact that USIMSCAR improves SBR for CBR classification using a real-world dataset.

Up to now, we have formalized the recommendation problem as a classification problem and shown the improvement of USIMSCAR over k-NN classifiers in terms of CA and MAE. In a certain context, it is also important to compare USIMSCAR and existing recommenders. Recommenders are usually classified as follows: content-based (CB) recommenders recommend items similar to the ones that the user has liked in the past, collaborative filtering (CF) recommenders recommend items that other users with similar preferences have liked in the past, and hybrid recommenders recommend items by combining the above two approaches. We see that USIMSCAR is also a unifying model realizing a hybrid recommendation. It differs from CF recommenders in that it exploits content information of items (movies) with rating information. It also differs from CB recommenders by using other users' ratings when building and exploiting association knowledge for rating classification. We compare USIMSCAR with two well-known hybrid recommenders: CLAYPOOL [20] and MELVILLE [21]. For CLAYPOOL, we first applied the CF method proposed by [20] to the training data. We then applied a CB method using IBk to the data. The ratings returned by these methods were combined by the equal-weighted average to produce a final rating. MELVILLE uses a CB method to convert a *sparse* user-ratings matrix UM into a *full* user-ratings matrix FUM. Given a user, a rating prediction is made for a new item using a CF method on the FUM. As the CB predictor, we used IBk. For the CF method, we implemented the algorithm in [21].

The comparison results are seen in Tables 7 and 8. As seen in Table 7, USIMSCAR outperforms the recommenders in all rating-scales in terms of CA. We discover that 50% of comparisons between USIMSCAR and the recommenders are deemed to be statistically significant through the Z-test at 95% confidence. As seen in Table 8, USIMSCAR also outperforms both recommenders in all rating-scales in terms of MAE. We discover that 50% of comparisons between USIMSCAR and the recommenders are also deemed to be statistically

Table 7. The classification results

Recommenders	Classification Accuracy (%)		
	RS(5)	RS(3)	RS(2)
CLAYPOOL	48.95 (0.11%)	77.97 (0.22%)	80.04 (1.59%)
MELVILLE	43.96 (5.10% ⋆)	73.57 (4.40% ⋆)	75.86 (5.99% ⋆)
USIMSCAR	**49.06**	**79.56**	**80.26**

Table 8. The MAE results

Recommenders	Predictive Accuracy (MAE)		
	RS(5)	RS(3)	RS(2)
CLAYPOOL	.6954 (2.29%)	.3102 (1.28%)	.1996 (0.22%)
MELVILLE	.7863 (11.38% $\star$)	.3579 (6.95% $\star$)	.2414 (4.40% $\star$)
USIMSCAR	**.6725**	**.2884**	**.1974**

significant by the paired t-test with 95% confidence. In summary, through all the experiments, we have demonstrated the validity and soundness of our proposed USIMSCAR approach.

8 Conclusion and Future Work

This paper presented a novel retrieval strategy USIMSCAR that can be used in retrieving useful cases for the target problem. First, we proposed an approach for extracting and representing association knowledge that represents strongly evident, interesting associations between known problem features and solutions shared by a large number of cases. We proposed that this knowledge is encoded via soft-matching class association rules (scars) using association analysis techniques. We proposed USIMSCAR that leverages useful cases and rules, with respect to the target problem, quantified by using both similarity and association knowledge. This idea to leveraging the combined knowledge during CBR retrieval clearly distinguishes USIMSCAR from SBR as well as existing retrieval strategies developed in the CBR research community. We validated the improvement of USIMSCAR over well-known k-NN approaches for implementing SBR through experiments using the Yahoo! Webscope Movie dataset. The experimental results showed that USIMSCAR is an effective retrieval strategy for CBR.

In CBR, cases can also be represented by more complex structures, like object-oriented representation (OO) or hierarchical representation (HR) [2]. OO utilizes the data modeling approach of the OO paradigm, such as inheritance. In HR, a case is characterized through multiple levels of abstraction, and its attribute values can reference nonatomic cases [2]. To support these representations, USIMSCAR must address how to generate similarity knowledge and association knowledge. To address the former, one may use similarity measures proposed by [22] for OO data or HR data. To address the latter, one may integrate the soft-matching criterion and extended Apriori algorithms such as OR-FP [23] for OO data and DFMLA [24] for HR data.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 7, 39–59 (1994)
2. Lopez De Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision and retention in case-based reasoning. Knowl. Eng. Rev. 20, 215–240 (2005)

3. Smyth, B., Keane, M.T.: Adaptation-guided retrieval: questioning the similarity assumption in reasoning. Artif. Intell. 102, 249–293 (1998)
4. Stahl, A.: Learning of knowledge-intensive similarity measures in case-based reasoning. PhD thesis, Technical University of Kaiserslautern (2003)
5. Cercone, N., An, A., Chan, C.: Rule-induction and case-based reasoning: hybrid architectures appear advantageous. IEEE Trans. on Know. and Data Eng. 11, 166–174 (1999)
6. Park, Y.J., Kim, B.C., Chun, S.H.: New knowledge extraction technique using probability for case-based reasoning: application to medical diagnosis. Expert Systems 23, 2–20 (2006)
7. Castro, J.L., Navarro, M., Sánchez, J.M., Zurita, J.M.: Loss and gain functions for CBR retrieval. Inf. Sci. 179, 1738–1750 (2009)
8. Ahn, H., Kim, K.J.: Global optimization of case-based reasoning for breast cytology diagnosis. Expert Syst. Appl. 36, 724–734 (2009)
9. Bradley, K., Smyth, B.: Personalized information ordering: a case study in online recruitment. Knowledge-Based Systems 16, 269–275 (2003)
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)
11. Aamodt, A.: Knowledge-Intensive Case-Based Reasoning in Creek. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 793–850. Springer, Heidelberg (2004)
12. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD 1993, pp. 207–216. ACM (1993)
13. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the 4th KDD, pp. 443–447 (1998)
14. Nahm, U.Y., Mooney, R.J.: Mining soft-matching association rules. In: Proceedings of CIKM 2002, pp. 681–683 (2002)
15. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Comput. Surv. 38, 9 (2006)
16. Hu, T., Sung, S.Y., Xiong, H., Fu, Q.: Discovery of maximum length frequent itemsets. Inf. Sci. 178, 69–87 (2008)
17. Jurisica, I., Glasgow, J.: Case-Based Classification Using Similarity-Based Retrieval. In: International Conference on Tools with Artificial Intelligence, p. 410 (1996)
18. Cleary, J.G., Trigg, L.E.: K*: An Instance-based Learner Using an Entropic Distance Measure. In: Proceedings of the 12th ICML, pp. 108–114 (1995)
19. Richard, C.S.: Basic Statistical Analysis. Allyn & Bacon (2003)
20. Claypool, M., Gokhale, A., Miranda, T.: Combining content-based and collaborative filters in an online newspaper. In: Proceedings of ACM-SIGIR Workshop on Recommender Systems (1999)
21. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: AAAI 2002, pp. 187–192 (2002)
22. Bergmann, R., Stahl, A.: Similarity measures for object-oriented case representations. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 25–36. Springer, Heidelberg (1998)
23. Kuba, P., Popelinsky, L.: Mining frequent patterns in object-oriented data. In: Technical Report: Masaryk University Brno, Czech Republic (2005)
24. Pater, S.M., Popescu, D.E.: Market-Basket Problem Solved With Depth First Multi-Level Apriori Mining Algorithm. In: 3rd International Workshop on Soft Computing Applications SOFA 2009, pp. 133–138 (2009)