Analogical Trust Reasoning

Mozhgan Tavakolifard[1], Peter Herrmann[2], and Pinar Öztürk[3]

[1] Centre for Quantifiable Quality of Service in Communication Systems (Q2S),
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
mozhgan@q2s.ntnu.no

[2] Department of Telematics (ITEM), Norwegian University of Science and Technology
(NTNU), Trondheim, Norway
herrmann@item.ntnu.no

[3] Computer and Information Science Department (IDI), Norwegian University of Science and
Technology (NTNU), Trondheim, Norway
pinar@idi.ntnu.no

Abstract. Trust is situation-specific and the trust judgment problem with which
the truster is confronted might be, in some ways, similar but not identical to some
problems the truster has previously encountered. The truster then may draw infor-
mation from these past experiences useful for the current situation. We present a
knowledge-intensive and model-based case-based reasoning framework that sup-
ports the truster to infer such information. The suggested method augments the
typically sparse trust information by inferring the missing information from other
situational conditions, and can better support situation-aware trust management.
Our framework can be coupled with existing trust management models to make
them situation-aware. It uses the underlying model of trust management to trans-
fer trust information between situations. We validate the proposed framework for
Subjective Logic trust management model and evaluate it by conducting experi-
ments on a large real dataset.

1 Introduction

This paper presents a context management framework (CMF) that employs case-based
reasoning [19] to analyze the correlation between trust information among various situ-
ations and help to bootstrap in unanticipated situations using trust information available
from similar situations. The case-based reasoning technique is particularly useful for
tasks that are experience-intensive, that involve plausible (i.e. not sound) reasoning and
have incomplete rules to apply.

The fundamental principle of the case-based reasoning technique is similar to that
of the human analogical reasoning process which employs solutions of past problems
to solve current ones. The reasoning process is generally composed of three stages:
remembering, reusing, and learning. Remembering is the case-retrieval process, which
retrieves relevant and useful past cases. In the reusing step, the case-based reasoning
system applies the cases that have been retrieved to find an effective solution to the
current problem. Learning is the process of casebase enhancement. At the end of each
problem-solving session the new case and problem-solving experiences incorporated
into the casebase [15].

We present a universal mechanism (called CMF) that can be combined with existing trust management models (TMM) to extend their capabilities towards efficient modeling of the situation-aware trust by

- estimating the trust values based on similar situations, in unknown situations or for unknown trustees when there is no information available. Therefore, CMF can help TMM to bootstrap (Figure 1(a)).
- adjusting the output of TMM (trust value) based on the underlying situation, thus, providing situation-awareness for TMM (Figure 1(b)).

In our approach TMM is implemented using the Subjective Logic [12]. One of our main contributions is the extension of the Subjective Logic with a context-sensitive domain model.
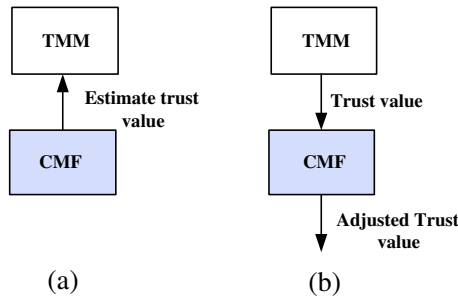
Fig. 1. Scope and interconnection of context management framework (CMF) and trust management model (TMM). a) Estimation of the trust value in unknown situations. b) Adjustment of the output of TMM (trust value) based on the underlying situation.

The rest of this paper is organized as follows: In section 2, we briefly explain the Subjective Logic as an example of the trust management model. Our proposed model for trust inference is described in section 3. Next in section 4, we present the evaluation plan and results. Section 5 provides an overview of the related research. Finally, conclusion and some ideas for future work are given in section 6.

2 Subjective Logic Trust Management Model

In this section, we briefly explain the Subjective Logic fundamentals and give reasons why it needs to be extended with a *situation* dimension. Subjective Logic [10] enables the representation of a specific belief calculus in which trust is expressed by a belief metric called opinion. An opinion is denoted by $\omega_B^A = (b, d, u, a)$ expressing the belief of a relying party A in the trustworthiness of another party B. The parameters b and d represent the belief respectively. disbelief in B's trustworthiness while u expresses the uncertainty in A's trust in B. All the three parameters are probability values between 0 and 1, and fulfill the constraint $b + d + u = 1$. The parameter a is called the base rate and determines how uncertainty contributes to the opinion's probability expected

value which is calculated as $E(\omega_x^A) = b + au$. The opinion space can be mapped into the interior of an equal-sided triangle, where the three parameters b, d, and u determine the position of the point in the triangle representing the opinion.

Based on the Subjective Logic, there are two different types of trust relations: *functional trust* (FT_B^A) and *referral trust* (RT_B^A). The former concerns A's direct trust in B performing a specific task ,while the latter concerns A's trust in B giving a recommendation about someone else doing a task. In other words, it is the trust in the ability to refer to a suitable third party. The simplest form of trust inference is trust transitivity which is widely discussed in literature [4, 7, 23]. That is, if A trusts B who trusts C, then A will also trusts in C. A valid transitive trust path requires that the last edge in the path represents functional trust and that all other edges in the path represents referral trust. Referral trust transitivity and parallel combination of trust paths are expressed as part of the Subjective Logic model (figure 2) [12].
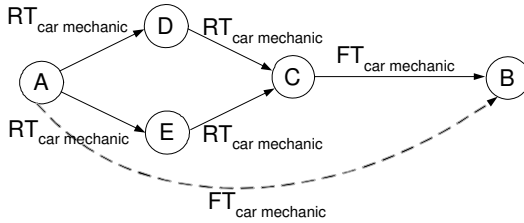
Fig. 2. Trust transitivity and parallel combination of trust paths. FT is functional trust and RT is referral trust.

The discounting operator ($\otimes$) [11] is used to derive trust from transitive trust paths, and the consensus operator ($\oplus$) allows to combine parallel transitive trust paths. The trust network in figure 2 can then be expressed as

$$FT_B^A = ((RT_D^A \otimes RT_C^D) \oplus (RT_E^A \otimes RT_C^E)) \otimes FT_B^C \qquad (1)$$

There are two reasons for extension of the Subjective Logic with situation representation. First, It has been shown [3] that trust is not always transitive in real life. For example, the fact that A trusts B to fix her car and B trusts C to look after his child does not imply that A trusts C for fixing the car, or for looking after her child. However, under certain semantic constraints, trust can be transitive and a trust referral system can be used to derive transitive trust. The semantic constraint in the Subjective Logic is that the subject of trust should be the same along the entire path, for example all trust subjects should be "to be a good car mechanic" (figure 2) or "looking after her child". On the other hand, this constraint is relaxed in our proposal by introducing the notion of situation. We suggest that trust situations along a transitive trust path can be different but similar to each other. For instance, trust situations can be "to be a good car mechanic" or "to be a good motor mechanic" (figure 3). In this way, we are able to use trust information from available similar situations (section 6 provides the details).
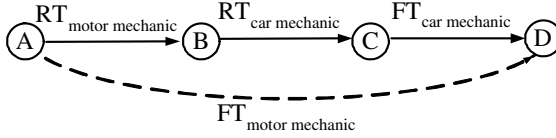
Fig. 3. Trust transferability among similar situations

Second, Jøsang introduces three different versions of the consensus operator (denoted by $\oplus$, $\underline{\oplus}$, $\tilde{\oplus}$ respectively) for fusion of independent, dependent, and partially dependent trust opinions [14]. If A and B have simultaneously observed the same event in the situation then their opinions are dependent. If A and B observed the same event during two partially overlapping situations then their opinions are partially dependent (e.g. A and B observed the same event of fire at the same time. A was in the place of fire, while B saw it on TV). Jøsang assumes that fraction of the overlapping observations is known and proposes formulas to estimate dependent and independent parts of the two observations to define the consensus operator of partially dependent opinions ($\tilde{\oplus}$). We propose to calculate the fraction of overlapping observations as the similarity measure between the two situations.

3   The Proposed Framework

We consider two approaches for the inference task among situations: rule-based inference and similarity-based reasoning, depicted respectively as case-based reasoner (CBR) and rule-based reasoner (RBR) modules in figure 4. The former provides the
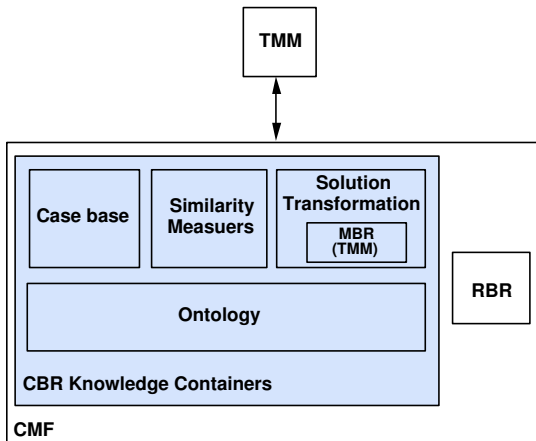

Fig. 4. Knowledge containers in case-based reasoner (CBR). TMM: trust management model, MBR: Model-based reasoner, RBR: rule-based reasoner, CMF: context management framework.

first role (Figure 1(a)), estimation of the trust value in unanticipated situations and the latter is responsible for the second role (Figure 1(b)) of CMF, adjustment of the trust values based on underlying situation. The gray box in figure 4 shows the focus of this paper.

3.1 Case-Based Reasoner Module

In the case-based reasoning approach, knowledge is distributed among the four knowledge containers: ontology, casebase, similarity measures, and solution transformation.

- *Ontology*: We represent the situations in the pertinant domain in form of an ontology. A situation consists of set of *contexts* which are captured as nodes of the ontology. Figure 5 depicts the ontology related to user-movie ratings. In this example, a situation has two main contexts: *User* and *Movie*. Demographic information for the users (age, occupation, sex, and zip code) are *local contexts* for the *User* context and movie genres are *local contexts* for the *Movie* context.
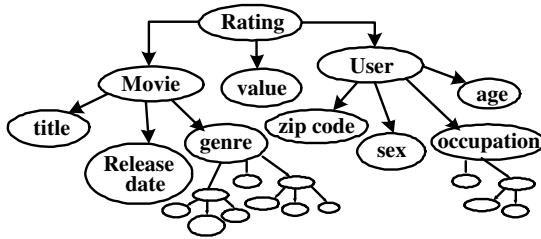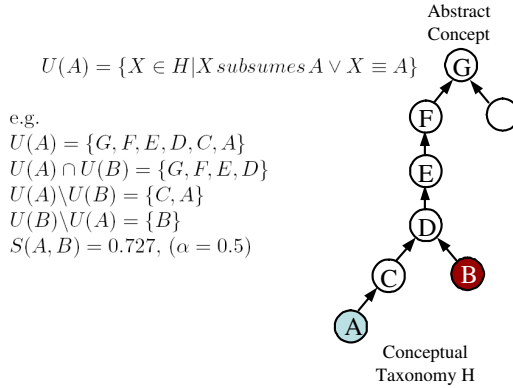
Fig. 5. The ontology example for user-movie ratings

- *Casebase*: The characterizations of the previous experiences and the recommendations (trust information including truster, trustee, trust value, and situation) are stored as elements of cases in the casebase. Cases are represented as attribute-value pairs.
- *Similarity*[1]: The similarity between situations is a weighted sum of the similarity between their contexts. Similarity between contexts, in turn, are computed as the wighted sum of the similarity between the underlying local contexts. According to the Tverskys formula [30], the similarity between two concepts A and B can be determined in the following way:

$$S(A,B) = \frac{|U(A) \cap U(B)|}{|U(A) \cap U(B)| + \alpha |U(A) \setminus U(B)| + (1 - \alpha) |U(B) \setminus U(A)|} \quad (2)$$

$U(A)$ and $U(B)$ are the sets of properties of concepts A and B, respectively. The function U takes into account the depth of compared concepts in the ontology hierarchy. α is a value in the range $[0, 0.5]$. The value of 0 implies that the differences of A with respect to B are not sufficient to conclude that they are similar, and the value

$$U(A) = \{X \in H | X \, subsumes \, A \vee X \equiv A\}$$

e.g.
$U(A) = \{G, F, E, D, C, A\}$
$U(A) \cap U(B) = \{G, F, E, D\}$
$U(A) \backslash U(B) = \{C, A\}$
$U(B) \backslash U(A) = \{B\}$
$S(A, B) = 0.727, (\alpha = 0.5)$

Fig. 6. Relations taxonomy

of 0.5 means that the differences are necessary and sufficient to conclude such an assumption. Figure 6 illustrates an example of the similarity calculation.

In our approach, equation (2) is used to compare the attributes with each other, while the comparison between the *values* of an attribute is performed using the following general comparasion guidelines:

– *Categorical*: values in the same category are similar (e.g., weather).
– *Continuous*: closer values are alike (e.g., time).
– *Hierarchical*: values in the same hierarchy are similar (e.g., location).

Attributes which do not have these characteristics may require a custom comparator to be defined for them.

• *Solution transformation*: The model-based reasoner (MBR) is responsible for adaptation or transformation of a solution (trust value) from previous experiences to the current problem of trust judgment. It uses TMM to estimate trust value for the current situation based on trust values of the similar situations (see figure 4). In section 3.2.1, we consider the Subjective Logic model as TMM and provide details for the solution transformation module.

3.2 Processes

CMF is generally composed of three processes: Remembering, Reusing, and Learning.

• *Remembering*: The query (the current trust assessment question) is compared to cases (past trust assessment experiences) in the casebase and N most similar cases are retrieved (N nearest neighbors). This process uses the ontology to measure the similarity between the query and each case in the casebase.
• *Reusing*: A trust value is predicted for the query using the solution transformation module.
• *Learning*: A new case is built from the query and the predicted value and is added to the casebase for future uses.

---

[1] In [27] we provide a comprehensive set of similarity measurement algorithms.

In following, we explain the details for solution transformation module considering the Subjective Logic as TMM.

3.2.1 Solution Transformation in Case of the Subjective Logic

We explain the functionality of the model-based reasoner through extension of the Subjective Logic model as TMM. If A has functional trust in B in situation C_1, then A can infer its functional trust to B in situation C_2 which is a similar situation. For example, if A trusts B as a good car mechanic then A will probably trust B in repairing motorcycles since there is a large similarity between the domains of repairing cars and motorcycles.

Similarly to Jøsang's way to define opinions, we use triples to describe similarity which enables us to use the Subjective Logic operators.

Definition 1. *The similarity opinion $S_{C_1}^{C_2}$ from C_1 towards C_2 is the triple[2] (similarity, non-similarity, uncertainty) and fulfills the constraints that the sum of all three values is equal to 1. If $C_1 = C_2$, the similarity opinion is defined to be $(1,0,0)$. Otherwise, it is calculated based on the measure of similarity $(S(C_1, C2))$ between the two situations C_1 and C_2 and the depth of concepts in the ontology (see (2)):*

$$S_{C_1}^{C_2} = (\frac{S(C_1,C_2) \cdot UN(C_1,C_2)}{k + UN(C_1,C_2)}, \frac{(1 - S(C_1,C_2)) \cdot UN(C_1,C_2)}{k + UN(C_1,C_2)}, \frac{k}{k + UN(C_1,C_2)}) \quad (3)$$

Here, k is a constant and $UN(C_1,C_2) = |U(C_1) \cup U(C_2)|$ defining the number of properties in play at all. In general, the higher the similarity value is, the less uncertain we are, and the uncertainty will be lower as more details ($UN(C_1,C_2)$) are available in comparison of the two situations C_1 and C_2.

Our similarity opinion is a special form of referral trust. It reflects that the akin situations of C_1 and C_2 is a kind of recommendation (reminding) to A to treat in situations C_1 and C_2 similarly. Thus, we see the consensus operator $\otimes$ as the correct mechanism to combine the similarity opinion between C_1 and C_2 with the functional trust of A in B in order to infer the functional trust of A in B:

$$FT_{B,C_1}^{A} = S_{C_1}^{C_2} \otimes FT_{B,C_2}^{A} \quad (4)$$

$FT_{B,X}^{A}$ is extended notation for A's functional trust to B which considers the underlying situation X. The higher the similarity between C_1 and C_2 is, the closer the trust of A to B in situation C_1 will be equal to that of between A and B in situation C_2. The lower this similarity is, the more uncertain A will be about whether to trust B or not in the second situation.

The same conversion formula can be used for Referral Trust.

$$RT_{B,C_1}^{A} = S_{C_1}^{C_2} \otimes RT_{B,C_2}^{A} \quad (5)$$

---

[2] This metric is inferred from a metric for the trust value computation [13] by Jøsang and Knapskog.

4   Evaluation

We chose MovieLens data[3] in view of the fact that we needed a context-enriched data to evaluate our work. The MovieLens data has been collected by the GroupLens Research Project at the University of Minnesota[4]. The data consists of 100,000 ratings from 943 users on 1682 movies with every user having at least 20 ratings and simple demographic information for the users is included. Figure 5 depicts the ontology which corresponds to the MovieLens data.

User attributes are age, sex and 19 occupation categories[5], zipcode, and movie attributes are 19 film genres[6]. Much richer movie content can be obtained from the Internet Movie Database (IMDB)[7]. We consider user and movie concepts as contexts and user and movie attributes as local contexts to form the situation for each rating.

4.1   Data Setup

There are 5 datasets which are 80%/20% splits of the data into training and test data (training set of 80,000 ratings, and the test set of 20,000 ratings). Each of these datasets have disjoint test sets; this is for 5 fold cross validation (where we repeat our experiment with each training and test set and average the results). The test sets are used as references for the accuracy of the predictions.

In the MovieLens data, rating values 1 and 2 represent negative ratings, 4 and 5 represent positive ratings, and 3 indicates ambivalence (we consider them as -2,-1,0,+1,+2). In order to convert these rating values to the Subjective Logic opinions (the triple $(b,d,u), b+d+u=1$) we can use the following conversion method:

$$b = \frac{\sum_{i=2}^{n} (i-1) \cdot f(i)}{c + (n-1) \cdot \sum_{i=1}^{n} f(i)}, \quad d = \frac{\sum_{i=1}^{n-1} (n-i) \cdot f(i)}{c + (n-1) \cdot \sum_{i=1}^{n} f(i)}, \quad u = \frac{c}{c + (n-1) \cdot \sum_{i=1}^{n} f(i)} \quad (6)$$

where the number of ratings at level i is described by function $f(i)$ and c is a constant.

4.2   Experimental Setup

The casebase is built up from the ratings in the training set. Each case is composed of four parts: user identifier, movie identifier, rating value, and situation including user and movie information. Ratings in the test set forms queries to CMF and each query is composed of three parts: user identifier, movie identifier, and the situation (the rating

---

[3] http://www.grouplens.org/node/73

[4] http://www.cs.umn.edu/Research/GroupLens/data/

[5] Occupation list: administrator, artist, doctor, educator, engineer, entertainment, executive, healthcare, homemaker, lawyer, librarian, marketing, none, other, programmer, retired, salesman, scientist, student, technician, writer.

[6] Film genres: unknown, action, adventure, animation, children, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war, western.

[7] http://us.imdb.com

value is removed). The rating value in the query is predicted by CMF using the casebase, and then consequently compared with the removed value in the test set.

Four types of evaluation criteria are used in this paper:

- Coverage: measure of the percentage of movies in the test dataset that can be predicted.
- FCP: fraction of correct predictions.
- MAE (Mean Absolute Error) : average of the prediction error (difference between probability expected values of predicted and real opinions) over all queries.
- RMSE (root mean squared error) : root mean of the average of the squared prediction error. RMSE tends to emphasize large errors.

The evaluation is described as a pseudo-code in algorithm 1. First, the casebase and the set of queries are built from training and test sets, respectively. Second, the *Remember* procedure is called for each query computes the similarity between each case in the casebase and the query. Cases with a similarity less than a threshold are ignored and the ten most similar cases among the remainings are retrieved. Next, by calling the *Reuse* procedure, a rating value is predicted for the query (R_q) based on the rating values of the retrieved cases ($R_i, i = 1..10$) and their similarity measures (S_i) which are calculated by the *Similarity* procedure.

$$R_q = (S_1 \otimes R_1) \oplus (S_2 \otimes R_2) \oplus \ldots \oplus (S_{10} \otimes R_{10}) \tag{7}$$

Then, a new case is built which contains user and movie information of the query and the predicted rating value is added to the casebase by calling the *Learn* procedure. The predicted ratings form the *predicted set*. Finally, the *test* and *predicted* sets are compared according to the four metrics (Coverage, FCP, MAE, and RSME) by calling the *Evaluate* procedure.

The *Similarity* procedure (see algorithm 2) calculates weighted average of similarity measures of local contexts (age, sex, occupation, and zipcode for users and genres for movies) to determine the similarity between situations. In our implementation these weights are 0.2, 0.15, 0.1, 0.05, 0.5 respectively and are determined based on the fact that how much the local context can affect the rating decision. The comparator for each local context are:

- Age: Closer values are more similar.
- Sex: The similarity value is 1 for identical sex values and 0 otherwise.
- Occupation: The similarity is calculated according to (2) for similarity measurement on the ontology.
- Zipcode: ZIP codes are numbered with the first digit representing a certain group of U.S. states, the second and third digits together representing a region in that group (or perhaps a large city) and the fourth and fifth digits representing a group of delivery addresses within that region. We assign similarity values of 1, 0.75, 0.5 to the same delivery address, region, and state group respectively.
- Movie genre: The similarity is calculated using (2) to measure similarity on the ontology.

Our baseline is the Pearson algorithm [17] which relies on Pearson correlation co-efficient to produce a correlation metric between users. This correlation is then used to weigh the rating of each relevant user. The Pearson correlation between users A and B is defined as:

$$P_{A,B} = \frac{\sum_{i=1}^{m} (R_{A,i} - \bar{R}_A) \times (R_{B,i} - \bar{R}_B)}{\sigma_A \times \sigma_B} \tag{8}$$

Algorithm 1. CONTEXT MANAGEMENT FRAMEWORK(*test_set*, *training_set*)

---

main
  global *casebase*, *similarity*
  comment: Build "casebase" from the training set and "queries" from the test set

  similarity[1..*size*(*casebase*)] ← 0
  comment: "similarity" array stores similarity measures between the query and the cases

  for each *query* ∈ *queries*
    do $\begin{cases} neighbors \leftarrow \text{REMEMBER}(query, casebase) \\ predicted\_rating \leftarrow \text{REUSE}(neighbors) \\ \text{LEARN}(query, predicted\_rating) \\ predicted\_set \leftarrow predicted\_set \cup predicted\_rating \end{cases}$
  EVALUATE(*test_set*, *predicted_set*)

  procedure REMEMBER(*query*)
  for each *case* ∈ *casebase*
    do $\begin{cases} sim \leftarrow \text{SIMILARITY}(query, case) \\ \textbf{if } sim >= THRESHOLD \\ \quad \textbf{then } similarity[case] \leftarrow sim \end{cases}$
  return (ten most similar cases)

  procedure REUSE(*neighbors*)
  predicated_opinion ← (0, 0, 1)
  for each *ncase* ∈ *neighbors*
    do $\begin{cases} similarity\_opinion \leftarrow (similarity[ncase], 0, 1 - similarity[ncase]) \\ new\_opinion \leftarrow similarity\_opinion \otimes ncase.rating \\ predicted\_opinion \leftarrow predicted\_opinion \oplus new\_opinion \end{cases}$
  return (*predicted_opinion*)

  procedure LEARN(*query*, *predicted_rating*)
  new_case ← *query.user* ∪ *query.movie* ∪ *predicted_rating*
  casebase ← *casebase* ∪ *new_case*

  procedure EVALUATE(*test_set*, *predicted_set*)
  coverage ← fraction of predicted ratings
  fcp ← fraction of correct predictions
  mae ← mean absolute error of predictions
  rmse ← root mean squared error of predictions
  output (*coverage*, *fcp*, *mae*, *rmse*)

Algorithm 2. SIMILARITY(*query*, *case*)

---

procedure SIMILARITY(*query*, *case*)

$user_q \leftarrow query.user$

$user_c \leftarrow case.user$

$age\_sim \leftarrow 1 - \frac{age_q - age_c}{age_{max} - age_{min}}$

if $sex_q == sex_c$

  then $sex\_sim \leftarrow 1$

  else $sex\_sim \leftarrow 0$

$occupation\_sim \leftarrow$ ONTOLOGYSIM($occupation_q, occupation_c$)

comment: "OntologySim" calculates contextual similarity according to (2)

if $zipcode_q(1) == zipcode_c(1)$

  then $\begin{cases} \textbf{if } zipcode_q(2,3) == zipcode_c(2,3) \\ \quad \textbf{then } \begin{cases} \textbf{if } zipcode_q(4,5) == zipcode_c(4,5) \\ \quad \textbf{then } \begin{cases} zipcode\_sim \leftarrow 1 \\ \textbf{comment: the same delivery address} \end{cases} \\ \quad \textbf{else } \begin{cases} zipcode\_sim \leftarrow 0.75 \\ \textbf{comment: the same region} \end{cases} \end{cases} \\ \quad \textbf{else } \begin{cases} zipcode\_sim \leftarrow 0.5 \\ \textbf{comment: the same state group} \end{cases} \end{cases}$

  else $zipcode\_sim \leftarrow 0$

$movie\_sim \leftarrow$ ONTOLOGYSIM($movie_q.genre, movie_c.genre$)

$total\_sim \leftarrow 0.2 \cdot age\_sim + 0.15 \cdot sex\_sim + 0.1 \cdot occupation\_sim$

$+ 0.05 \cdot zipcode\_sim + 0.5 \cdot movie\_sim$

return ($total\_sim$)

---

where m is the number of movies that both users rated. $R_{A,i}$ is the rating, user A gave to movie i. $\bar{R}_A$ is the average rating user A gave to all movies, and σ_A is the standard deviation of those ratings. Once the Pearson correlation between a user and all other users is obtained, the predicted movie rating is calculated as:

$$R_{A,i} = \bar{R}_A + \frac{\sum_{U=1}^{n} (R_{U,i} - \bar{R}_U) \times P_{A,U}}{\sum_{u=1}^{n} |P_{A,U}|} \tag{9}$$

Use of the Pearson correlation coefficient is quite common in the field of collaborative filtering, and results obtained with this method will be used to gauge the performance of other algorithms. Moreover, the Pearson algorithm uses only the rating information while our method use situational information to do the prediction.

4.3 Discussion of the Obtained Results

In table 1, we present the final results of the evaluation. We start by commenting the row "Coverage". The coverage becomes an important issue on a very sparse dataset

Table 1. Final evaluation results

Metric	Dataset1	Dataset2	Dataset3	Dataset4	Dataset5	Average	Pearson CC
Coverage (%)	43.82	43.88	44.94	45.42	45.06	44.62	99.83
FCP	0.3629	0.3497	0.3299	0.3345	0.3417	0.3437	0.1993
MAE	0.1605	0.1600	0.1656	0.1648	0.1626	0.1627	0.3049
RMSE	0.2742	0.2717	0.2757	0.2739	0.2724	0.2736	0.3804

that contains a large portion of cold-start users since many trust values become hardly predictable [18]. The results (*Coverage* ≈ 0.45%) indicate that our model is able to predicate approximately one rating from each two ratings. For the Pearson algorithm the coverage is not perfect merely because not all movies in the test dataset have a rating in the training dataset. The second important result is the fraction of correct predictions (FCP) is 0.34 which shows that from each 10 predicted ratings between 3 and 4 ratings are predicted with exact values. Further, the prediction errors (MAE and RMSE) for the other ratings that are not predicted exactly (between 6 and 7 ratings from each 10 predicted ratings) are small in comparison with the Pearson method (*MAE* ≈ 0.12 & *RMSE* ≈ 0.20).

All-in-all, the results of the evaluation lead to the expectation that our approach provides an improvement over the Pearson algorithm and this implies that situational information is useful in making predictions.

5   Related Research

CMF is a *knowledge-intensive CBR* which is designed to extend situational inference capabilities of *trust* management models. More precisely, the aim is to reuse the available trust information (direct experiences and recommendations) in similar situations for the current problem and we use semantic (ontology-based) similarity measures. Although CBR techniques are extensively used for recommender systems [1, 24] and there are some works which use CBR to build more trust through providing explanations [16, 21, 22], to the best of our knowledge this proposal is quite new. In this section, we briefly explain the related researches which are based on context-aware trust management and thus more closely resemble our goal.

According to the literature, the extension of a trust model with context representation can reduce complexity in the management of trust relationships [20], improve the recommendation process [20], help to infer trust information in context hierarchies [9], improve performance [25], help to learn policies/norms at runtime [25, 29], and provide protection against changes of identity and first time offenders [25]. Context related information has been represented as Context-aware domains [20], Intensional Programming [31], Multi-dimensional goals [8], Clustering [25], and Ontologies [29].

[26] provides a survey of different approaches to model context for ubiquitous computing. In particular, numerous approaches are reviewed, classified relative to their core elements and evaluated with respect to their appropriateness for ubiquitous computing.

The authors conclude that the most promising assets for context modeling of ubiquitous computing environments can be found in the ontology category in comparison with other approaches like key-value models, mark-up scheme models, graphical models, object-oriented models, and logic based models. This selection is based on the six requirements dominant in pervasive environments: distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality, and applicability to existing environments.

We present a state-of-the-art survey of context representation for trust management in [28]. In the rest of this section ontology-based approaches to this problem are examined in more details.

Golbeck et al. [6] propose an ontology for trust. In [5] the authors consider a model using context-specific reputation by assigning numeric ratings to different types of connections based on context of the analysis. In [29] rules to describe how certain context-sensitive information (trust factors) reduces or enhances the trust value have been specified for this trust ontology.

In [29] contextual information (i.e., context attributes) is used to adjust the output of a trust determination process. Each attribute can adjust the trust value positively or negatively according to a specified weight. As an illustration, if t is the trust value and ω is the weight of the context property then the adjusting function can be t^{ω} for decrease or $\sqrt[\omega]{t}$ for increase. A context ontology connects the context attributes with each other in an appropriate manner, enabling the utilization of context attributes which do not exactly match the query, but are "close enough" to it.

In [2], cases where a truster does not have enough information to produce a trust value for a given task, but she knows instead the previous partner behavior performing similar tasks, are considered. This model estimates trust using the information about similar tasks. The similarity between two tasks is obtained from the comparison of the task attributes.

6   Conclusion and Future Directions

To sum up, we propose a framework based on the case-based reasoning paradigm and the representation of deep knowledge to make existing trust management models situation-aware. This framework has been validated for the Subjective Logic trust management model as an example and evaluated using a real large-scale dataset. It can also be considered as an inference mechanism which deals with the sparsity and cold-start problems of a web of trust.

The original Subjective Logic can be applied to determine transitivity only if the subject of the trust relations along the entire path is the same. However, trust relations with the same subject are not always available. Our proposal opens up the possibility to draw transitivity also when the subject (situation) of the available trust relations are not the same but are similar. First, the trust relations with similar situations with the current problem are retrieved from the casebase using the ontology and the similarity measurement algorithm (remembering past similar trust experiences). Next, they are converted (using (4) and (5)) to equivalent trust relations in the current problem by solution transformation module (reusing the trust information from the past similar trust

experiences). Then, the transitive trust path is formed and final trust is calculated according to the Subjective Logic (1). Solution of the current problem is stored as a new case in the casebase (the learning process of CBR).

In the future, we aim to add a Risk Management Module to this framework. Risk evaluation becomes important in inferring trust values among situations especially when the trustworthiness of some principal is completely unknown and no recommendation information is available. The intuitive idea behind such a risk assessment can be to look up the in the casebase to see if there are any similar previous interactions, i.e., if we have previously encountered an entity with similar trust attributes and similar risk attributes in the same situation. The ontology part should be able to describe the level of situational risk, whereby the higher the risk of negative outcome, the higher the level of precision that must be captured.

References

1. Aguzzoli, S., Avesani, P., Massa, P.: Collaborative Case-Based Recommender Systems. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS, vol. 2416, pp. 460–474. Springer, Heidelberg (2002)
2. Caballero, A., Botía, J.A., Gómez-Skarmeta, A.F.: On the Behaviour of the TRSIM Model for Trust and Reputation. In: Petta, P., Müller, J.P., Klusch, M., Georgeff, M. (eds.) MATES 2007. LNCS, vol. 4687, pp. 182–193. Springer, Heidelberg (2007)
3. Christianson, B., Harbison, W.S.: Why Isn't Trust Transitive? In: Lomas, M. (ed.) Security Protocols 1996. LNCS, vol. 1189, pp. 171–176. Springer, Heidelberg (1997)
4. Ding, L., Kolari, P., Ganjugunte, S., Finin, T., Joshi, A.: Modeling and Evaluating Trust Network Inference. Technical report, Maryland Univ. Baltimore Dept. of Computer Science and Electrical Engineering (2005)
5. Golbeck, J., Hendler, J.: Inferring Reputation on the Semantic Web. In: Proceedings of the 13th International World Wide Web Conference (2004)
6. Golbeck, J., Parsia, B., Hendler, J.: Trust Networks on the Semantic Web. In: Klusch, M., Omicini, A., Ossowski, S., Laamanen, H. (eds.) CIA 2003. LNCS, vol. 2782, pp. 238–249. Springer, Heidelberg (2003)
7. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proceedings of the 13th international conference on World Wide Web, pp. 403–412. ACM Press, New York (2004)
8. Gujral, N., DeAngelis, D., Fullam, K.K., Barber, K.S.: Modeling Multi-Dimensional Trust. In: The Proceedings of the Workshop on Trust in Agent Societies, pp. 8–12 (2006)
9. Holtmanns, S., Yan, Z.: Context-Aware Adaptive Trust
10. Jøsang, A.: A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
11. Jøsang, A.: The consensus operator for combining beliefs. Artificial Intelligence 141(1-2), 157–170 (2002)
12. Jøsang, A., Hayward, R., Pope, S.: Trust network analysis with subjective logic. In: Proceedings of the 29th Australasian Computer Science Conference, vol. 48, pp. 85–94. Australian Computer Society (2006)
13. Jøsang, A., Knapskog, S.J.: A metric for trusted systems. In: Proceedings of the 21st National Security Conference, NSA (1998)
14. Josang, A., Marsh, S., Pope, S.: Exploring Different Types of Trust Propagation. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 179–192. Springer, Heidelberg (2006)

15. Jung, C., Han, I., Suh, B.: Risk Analysis for Electronic Commerce Using Case-Based Reasoning. Int. J. Intell. Sys. Acc. Fin. Mgmt. 8, 61–73 (1999)
16. Leake, D.B.: CBR in Context: The Present and Future. Case-Based Reasoning: Experiences, Lessons, and Future Directions, 3–30 (1996)
17. Massa, P., Avesani, P.: Trust-Aware Collaborative Filtering for Recommender Systems. In: Meersman, R., Tari, Z. (eds.) OTM 2004. LNCS, vol. 3290, pp. 492–508. Springer, Heidelberg (2004)
18. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proceedings of the 2007 ACM conference on Recommender systems, pp. 17–24. ACM Press, New York (2007)
19. Morris, B.W.: SCAN: a case-based reasoning model for generating information system control recommendations. International Journal of Intelligent Systems in Accounting, Finance and Management 3(1), 47–63 (1994)
20. Neisse, R., Wegdam, M., van Sinderen, M., Lenzini, G.: Trust Management Model and Architecture for Context-Aware Service Platforms. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part II. LNCS, vol. 4804, pp. 1803–1820. Springer, Heidelberg (2007)
21. Pu, P., Chen, L.: Trust building with explanation interfaces. In: Proceedings of the 11th international conference on Intelligent User Interfaces, pp. 93–100. ACM, New York (2006)
22. Pu, P., Chen, L.: Trust-inspiring explanation interfaces for recommender systems. Knowledge-Based Systems 20(6), 542–556 (2007)
23. Quercia, D., Hailes, S., Capra, L.: Lightweight Distributed Trust Propagation. In: Seventh IEEE International Conference on Data Mining, ICDM 2007, pp. 282–291 (2007)
24. Recio-García, J.A., Bridge, D.G., Díaz-Agudo, B., González-Calero, P.A.: CBR for CBR: A case-based template recommender system for building case-based systems. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS, vol. 5239, pp. 459–473. Springer, Heidelberg (2008)
25. Rehak, M., Pechoucek, M.: Trust modeling with context representation and generalized identities. In: Klusch, M., Hindriks, K.V., Papazoglou, M.P., Sterling, L. (eds.) CIA 2007. LNCS, vol. 4676, pp. 298–312. Springer, Heidelberg (2007)
26. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp (2004)
27. Tavakolifard, M., Knapskog, S., Herrmann, P.: Cross-Situation Trust Reasoning. In: Proceedings of The Workshop on Web Personalization, Reputation and Recommender Systems (WPRRS 2008). IEEE Computer Society Press, Los Alamitos (2008)
28. Tavakolifard, M., Knapskog, S., Herrmann, P.: Trust Transferability Among Similar Contexts. In: Proceedings of The 4th ACM International Workshop on QoS and Security for Wireless and Mobile Networks (Q2SWinet 2008). ACM, New York (2008)
29. Toivonen, S., Lenzini, G., Uusitalo, I.: Context-aware trust evaluation functions for dynamic reconfigurable systems. In: Proceedings of the Models of Trust for the Web Workshop (MTW 2006), held in conjunction with the 15th International World Wide Web Conference (WWW 2006), May 2006, vol. 22 (2006)
30. Tversky, A., et al.: Features of similarity. Psychological Review 84(4), 327–352 (1977)
31. Wan, K., Alagar, V.: An Intensional Functional Model of Trust. In: Proceedings of IFIPTM 2008 - Joint iTrust and PST Conferences on Privacy, Trust Management and Security, pp. 69–85. Springer, Boston (2008)