A Fuzzy Logic Approach to Case Matching and Retrieval suitable to SQL Implementation

Luigi Portinale, Stefania Montani
Dipartimento di Informatica
Universita’ del Piemonte Orientale ”A. Avogadro”, Alessandria, Italy
{portinal,stefania}@di.unipmn.it

Abstract

The aim of this paper is to formally introduce a fuzzy logic based notion of acceptance and similarity among case features, for case matching and retrieval. In particular, we present an approach where local acceptance relative to a feature can be expressed through fuzzy distributions on its domain, abstracting the actual values to linguistic terms. Furthermore, global acceptance is completely grounded on fuzzy logic, by means of the usual combinations of local distributions through specific defined norms. We propose a retrieval architecture, based on the above notions and implemented through a fuzzy extension of SQL.

1 Introduction

Case retrieval and matching algorithms usually focus on implementing Nearest-Neighbor (NN) techniques; a case is retrieved by a weighted combination of local distance metrics relative to individual features between the case and the target (query) one [7]. A k-NN case retrieval algorithm will then return the k closest cases to the target one; the underlying assumption is that the similarity between two cases is the inverse of their distance, so that the k cases closest to the target are the k most similar ones. In several AI subfields different from CBR, the problem of approximate matching (so crucial to CBR), has been traditionally dealt with the use of fuzzy techniques. We can transpose such approaches to CBR by considering the definition of a fuzzy linguistic term over a case feature as a “similarity dimension” over which to compare the feature’s values; in fact, a linguistic variable over the domain of a feature can be exploited to characterize the match acceptance (and so the retrieval) of cases having similar (in the fuzzy sense) values for that particular feature. The underlying idea is that, if two feature values are “close” in membership with respect to a given fuzzy set to be used as a reference context, then these values are similar with respect to such a context.

A second aspect related to standard case retrieval concerns the fact that, in order to implement a particular algorithm, suitable case structuring and case base organization have to be devised [7]. This may be a further burden in the construction of a CBR system, especially if data concerning cases are already available in standard relational databases, as in many applications. A limited effort has been made, in order to define distance-based approaches directly exploiting SQL [17, 16]. On the contrary, in the last years, we assisted to a growing interest for the definition of fuzzy extensions to standard SQL [9, 11, 13, 2], leading to several proposals for the definition of an SQL-like language able to deal with fuzzy conditions for the selection of data. The availability of such tools can provide a direct implementation of a case retrieval system based on a fuzzy notion of acceptance, over a standard RDBMS.

The aim of this paper is to formally introduce, in a case retrieval system, a notion of case matching and acceptance based on a fuzzy logic characterization of similarity among case features; at the same time, we aim at proposing an actual retrieval architecture, based on the above notions and realized through a fuzzy extension of SQL directly implemented on a standard SQL engine. The rest of the paper is organized as follows: section 2 reviews the relationships between distance-based and fuzzy-based similarity; section 3 formalizes the case matching and retrieval process we are interested to model; section 4 deals with the implementation of the proposed framework. Finally, in section 5, a comparison with related works is shortly presented and the conclusions are drawn.

2 Modeling Case Acceptability with Fuzzy Logic

In the Nearest-Neighbor approach case similarity is defined as a dual function of case distance: the greater the distance, the smaller is the similarity and vice versa. Stan-
standard distance-based similarity, implicitly accept cases for retrieval, by means of the induction of acceptability regions around the target or query case [3]. Another approach could be that of starting from measures directly modeling acceptability. However, acceptability is a concept related to the query at hand, so an acceptability measure must be defined according to a reference query context. A concrete example of an acceptability measure can be a fuzzy membership function, defining linguistic terms over a feature. A value $x$ of a feature $f$ has an acceptability of $\mu_f^x(x)$, with respect to the reference $\mathcal{L}$ if $\mu_f^x(x)$ is the membership degree of $x$ in the fuzzy set $\mathcal{L}$. The standard notion of $\alpha$-cut of a fuzzy set (i.e. the set of elements having degree of membership in the fuzzy set greater than or equal to $\alpha$) can be used to define an acceptability region: all the elements in the $\alpha$-cut are accepted. Different acceptability thresholds (i.e. different $\alpha$s) give rise to different acceptability regions. In this case, the similarity dimension over which to compare the values of a feature is the linguistic variable or term on which the fuzzy set is defined.

Given two values $x$ and $y$ of a feature $f$ and a linguistic term $\mathcal{L}$ defined over $f$ via a fuzzy membership function $\mu_f^\mathcal{L}$, we can characterize the similarity of $x$ and $y$ with respect to $\mathcal{L}$ through the absolute difference in the membership in $\mathcal{L}$ of $x$ and $y$ (i.e. $|\mu_f^\mathcal{L}(x) - \mu_f^\mathcal{L}(y)|$). Smaller is the difference, greater is the similarity of $x$ and $y$ with respect to $\mathcal{L}$. This captures the intuition that values that belong to the fuzzy set $\mathcal{L}$ with a close degree are considered semantically similar with respect to $\mathcal{L}$ (see also [10]).

In fuzzy logic the reference context for characterizing acceptability is usually a linguistic term (i.e. a fuzzy set), there is a basic difference with respect to standard distance-based approaches to CBR, where such a reference is a feature value within the admissible range (or the “unknown” value if the feature is missing). As pointed out in [4], the relationship between a standard similarity function $SIM(u, v)$ and a fuzzy membership function can be established by considering the latter to be the membership of the fuzzy set $SIM$ modeling the notion of similarity between two values on the “universe of discourse” (i.e. the range of the considered case attribute), such that $\mu_{SIM}(u, v) = SIM(u, v)$. On the other hand, if one starts by considering a particular fuzzy set $\mathcal{L}$ over the universe (in practical a linguistic term defined over the attribute’s range), then the membership function of a value $u$ in $\mathcal{L}$ (i.e. $\mu_{\mathcal{L}}(u)$) can be interpreted as the similarity of $u$ with respect to any value $a$ which is focal for $\mathcal{L}$. A value $a$ is focal for a fuzzy set $\mathcal{L}$ if $\mu_{\mathcal{L}}(a) = 1$; this means that $a$ is a fully representative value for the concept represented by $\mathcal{L}$, so the membership of $u$ in $\mathcal{L}$ measures how close (i.e. similar) is $u$ to the considered concept or linguistic term.

Finally, in a fuzzy context, global acceptability is easily obtained by using fuzzy combination through suitable t-norms or t-conorms; this means that the standard framework of fuzzy logic provides a sound way of aggregating local information (at the feature level) into a global measure (at the case level). Despite that, it is worth noting that the problem of aggregation of local fuzzy measures deserves particular attention. In fact, as precisely pointed out in [4], there may be different interpretations of the actual meaning of a local (at the feature level) fuzzy acceptability function, depending on how we interpret a similarity value equal to 0 (0-similarity). If this constraints the elements to be definitely not equivalent, then a t-norm based combination should be used, while on the contrary (i.e. if a value of similarity equal to 0 just does not contribute to aggregation) a t-conorm would be the right choice. We will return on this aspect in the next section.

### 3 Fuzzy Case Matching and Retrieval

The problem of approximate matching is intrinsic to CBR, but the preferred way of approaching this problem does not usually exploit fuzzy logic as the main tool for performing such a matching process. Actually, this possibility has been evidenced quite early in the CBR literature [10], but the mainstream of subsequent works largely focused on distance-based approaches.

In this section, we aim at formalizing the case retrieval process, based on a notion of approximate matching directly grounded on fuzzy logic. As pointed out in the previous section, this does not totally depart from distance-based approaches, since there are strong relationships between the two categories; however, it provides a different starting point able to exploit the sound framework of fuzzy set theory and, as we will show in the next section, a powerful computational framework based on SQL for implementing the retrieval process. Let start by defining what is a case and which are the characteristics of case attributes.

**Definition 3.1** A storable case (or simply a case) $c$ is a set of pairs $(f, v)$ where $f$ is a feature and $v$ an admissible value for $f$. We denote as $Range(f)$ the set of admissible values for the feature $f$.

A case base is a set of storable cases $CB = \{c_i | c_i$ is a storable case$\}$.

**Definition 3.2** A feature $f$ is said to be nominal if $Range(f)$ is a finite set of elements with no ordering relation among them.

A feature $f$ is said to be linear if $Range(f)$ is an ordered set; in particular $f$ is discrete linear if $Range(f)$ is isomorphic to $X \subseteq N$ and $f$ is continuous linear if $Range(f)$ is isomorphic to $Y \subseteq \mathbb{R}$.

---

1In CBR, a case is usually structured in two different parts: the problem description and the problem solution; without lack of generality, we will assume that the $(feature, value)$ description is adopted for both parts.
Feature categorization is important in order to define the similarity measure associated with each single feature. In this paper, we will introduce feature similarities, by considering a set of linguistic terms as reference contexts for such similarities. These contexts formally take the form of fuzzy predicates. Depending on the category of the considered feature, three different types of fuzzy predicates are considered: (1) fuzzy predicates with continuous distribution: they can be defined on linear (both continuous and discrete) features; (2) fuzzy predicates with discrete distribution: corresponding to predicates defined through fuzzy sets with a point-based membership function; they can then be defined on both nominal as well as on discrete linear features.

Moreover, approximate matching may require the use of suitable operators for comparing values in the stored cases with those provided in the query. For this reason, we consider the possibility of defining fuzzy operators to relate features values. Also in this case we have different types of operators: (1) continuous operators: characterized by a continuous distribution function (e.g. the operator near over linear features, characterized by a trapezoidal, triangular or Gaussian-like curve centered at 0 and working on the difference of the operands); (2) discrete operators: defining a similarity relation characterized by a symmetric matrix of similarity degrees (e.g. the operator compatible over the attribute job defining to what degree a pair of different jobs are compatible). Given the above definitions, we can now characterize the fuzzy case retrieval process.

**Definition 3.3** A query case is a set of pairs \( q = \{ (f_1, v_1), \ldots, (f_n, v_n) \} \) such that each \( f_i \) is a feature and each \( v_i \) is either

1. a fuzzy linguistic term defined over \( \text{Range}(f_i) \) with membership function \( \mu_{v_i} : \text{Range}(f_i) \rightarrow [0, 1] \) or
2. an expression of the kind “\( \text{op}_i x_i \)” such that \( x_i \in \text{Range}(f_i) \) and \( \text{op}_i \) is a binary operator (either crisp or fuzzy) defined over \( f_i \).

The first point refers to the use of a generic fuzzy value, including those generated by fuzzy modifiers: indeed, if a fuzzy modifier is used (e.g. very, slightly, etc...), the result is just a modified membership function that can then be used instead of the original one. We can consider as a fuzzy modifier any boolean expression of fuzzy terms defined on the feature at hand: for instance, if the fuzzy term young and old are defined over the feature age, we could be interested in the fuzzy expression

\[
\text{age}, (\text{young OR (old AND NOT very old)})
\]

The fuzzy condition on feature age denotes a new fuzzy set \( F \) with relative membership computable from the original memberships for terms young and old; the whole expression can then be reduced to the more simple expression \( (\text{age}, F) \).

A special (and common) case for the second point is the situation when the considered operator is the equality operator (=). In this case the expression \( (f_i = v_i) \) simply denotes the standard condition of feature \( f_i \) assuming a crisp value \( v_i \).

The use of a fuzzy linguistic term in the query can be useful also in situations when the end user specifies an actual (crisp) value for a feature, but it requires to abstract that value. In such a case, the user can specify the equality on a crisp value (as in point 2) asking the system to fuzzify it, on the basis of the fuzzy sets or operators defined over the feature (see section 4). After the fuzzification provided by the system, the situation is then reduced to the first point of definition 3.3. For the sake of generality, let us indicate as \( P(f_i, v_i) \) the predicate related to the assignment of a condition \( v_i \) to feature \( f_i \); it can be a standard boolean predicate (in case \( v_i \) involves a crisp operator), or a fuzzy predicates if linguistic values are specified. For example \( P(\text{age} = 40) \) is true iff the feature age assumes the value 40; \( P(\text{age}, \text{young}) \) is true with degree \( \mu_{\text{young}}(x) \) if \( \mu_{\text{young}} \) is the membership function of the fuzzy set young defined over age and the feature age assumes values \( x \).

Notice that in our framework, each case in the case base (i.e. each storable case) does not have any fuzzy specification in its structure; fuzzy information is used only at the query level to retrieve stored cases on the basis of an approximate (fuzzy) match.

**Definition 3.4** Given a query case \( q = \{ (f_i, v_i) | 1 \leq i \leq n \} \), the retrieval condition induced by \( q \) is the fuzzy predicate \( RC_q \equiv \bigwedge_{i=1}^{n} P(f_i, v_i) \) where \( \bigwedge_{i=1}^{n} \) is a boolean expression involving all the predicates \( P(f_i, v_i) \).

The above definition (where \( n \) is the number of features specified in the query) allows one to determine which kind of role the matching features have to play in case retrieval. If for some features we ask to combine similarities in such a way that a 0-similarity (see end of section 2) fully contributes to non-equivalence, then a t-norm combination should be used, resulting in a predicate composition through the AND (\( \wedge \)) connective. If on the contrary, a 0-similarity does not contribute at all to non-equivalence, then a t-conorm should be the choice and the predicate composition should be performed through OR (\( \vee \)) connective. In case this two different kinds of concept should be combined, then the corresponding AND/OR formula can be adopted (see [4] for the details). In the following, we will concentrate on the case when \( RC_q \equiv \bigwedge_{i=1}^{n} P(f_i, v_i) \); this is the most common situation in CBR systems, where the features’ requirements explicitly stated by the user (\( P(f_i, v_i) \))
in the above definition) are actually interpreted as (possibly fuzzy) constraints that must simultaneously hold.

Definition 3.5 Given a storable case \( c \), a query case \( q \) and the retrieval condition \( RC_q \), the matching degree of \( c \) to \( q \) is \( \alpha(c, q) = \mu_{RC_q}(c) \)

Definition 3.6 Given a case base \( CB \), a query case \( q \) and a threshold \( \lambda \in [0, 1] \), the retrieval set of \( q \) with respect to \( CB \) and \( \lambda \) is the set \( RS(q, CB, \lambda) = \{ c_i \in CB | \alpha(c_i, q) \geq \lambda \} \)

The problem of finding the stored cases that best match the query, is then reduced to that of finding the set of cases satisfying, to an acceptable degree of match (represented by \( \lambda \)), the conditions specified in the query itself. The present framework fully exploit fuzzy logic for: (1) modeling the similarity of the case features with respect to different approximate concepts; (2) modeling the retrieval conditions as well as the acceptability of the retrieved cases. Furthermore, as mentioned in the introduction, a fuzzy characterization of case matching can be directly captured in SQL based tools. In fact, several approaches have been proposed in the literature, in order to deal with fuzzy information in standard databases. Among these approaches, those relevant to our framework are the ones proposing fuzzy querying on a standard relational database [2, 12, 11]. As already described in [14], a framework like the one described in section 2 and 3 can be easily implemented by exploiting the derivation principle [2], a methodology able to derive, from a fuzzy SQL query an equivalent standard SQL query retrieving the desired \( \lambda \)-cut. In particular, we can consider the following syntax:

\[
\text{SELECT} \ (\lambda) \ \ A \ \ \text{FROM} \ \ R \ \ \text{WHERE} \ \ fc
\]

which meaning is that a set of tuples with attribute set \( A \), from relation set \( R \), satisfying the condition \( fc \) with degree \( \mu \geq \lambda \) is returned.

4 Implementing Fuzzy Case Retrieval

In order to make effective our fuzzy case retrieval approach on top of a RDBMS we have defined the following implementation framework:

- a query case is defined by specifying conditions on the values for a set of features as indicated in definition 3.3.
- retrieval takes place, after specifying an acceptability threshold \( \lambda \), by generating a fuzzy SQL query on the case base with threshold \( \lambda \).

Concerning the query case specification, a particular attention must be paid when an expression of the type \( (f = v) \) is specified. This may be dealt with in two different ways, depending on the user intentions: (1) using the crisp condition \( f = v \), stating that the feature \( f \) must have value \( v \); (2) transforming the expression \( (f = v) \) into the expression \( (f, \mathcal{F}) \) where \( \mathcal{F} \) is the fuzzy set resulting from the fuzzification of the value \( v \) of the feature \( f \).

In the last case, fuzzification may take place in different ways. If the feature \( f \) is linear, the fuzzy expression \( f \text{ near}_\epsilon v \) can be generated, with \( \text{near}_\epsilon \) being a fuzzy operator modeling the proximity of values for the feature \( f \). This fuzzification strategy requires the knowledge engineer to specify such proximity operators for the features potentially subject to fuzzification. In case \( f \) is a nominal attribute, a fuzzification method can be defined by considering the membership \( \mu_1(v) \) with respect to a given fuzzy set \( F_1 \) defined on \( \text{Range}(f) \). In that case, a new fuzzy set \( F_2 \) with membership distribution \( \mu_2 \) can be defined such that \( \mu_2(v') = 1 \) for every \( v' \) such that \( \mu_1(v') = \mu_1(v) \) and by scaling proportionally the membership function for other values. Moreover, if a fuzzy operator \( \theta \) is defined on attribute \( f \), the expression \( (f = v) \) can be fuzzified by transforming the crisp operator \( = \) into \( \theta \).

Finally, in [10] another simple kind of fuzzification is proposed (applicable to both linear and nominal features); this fuzzification takes into account the level of acceptability \( \lambda \) specified for the retrieval; it simply transforms the expression \( (f = v) \) into the expression \( (f, \mathcal{F}_1) \) for each \( \mathcal{F}_1 \) such that \( \mu_{\mathcal{F}_1}(v) \geq \lambda \).

Once a query case \( q \) and an acceptability threshold \( \lambda \) have been specified on the case base \( CB \), a fuzzy SQL query can then be generated as follows:

\[
\text{SELECT} \ (\lambda) \ * \ \text{FROM} \ \ CB \ \ \text{WHERE} \ \ RC_q
\]

Retrieved cases will then be obtained as the tuples of the result table obtained from the query. Of course, if one is interested in just a subset \( A \) of the case features, the target list of the query will be \( A \) instead of *.

5 Conclusions and Related Works

We have presented an approach where local acceptance and similarity relative to a feature can be expressed through fuzzy distributions on its domain, abstracting the actual values to linguistic terms. Global acceptance and similarity are then completely defined in fuzzy terms, by means of the
usual combinations of local distributions through specific defined norms and so it is completely grounded on fuzzy logic. In [7] it is argued that fuzzy logic is really relevant in both case representation and in case retrieval, where “fuzzy knowledge representation” and “fuzzy matching” methods can provide the suitable tools. Also Yager in [19] argues for a unified view of CBR and fuzzy reasoning systems and works in [4, 8] are clearly steps in this direction. Despite that, the attempt of building CBR systems directly exploiting fuzzy logic for retrieval has not received as much attention as the definition of distance-based approaches, which are by now much more popular in the CBR community.

Remarkable exceptions, where the role of fuzzy logic in CBR is emphasized, are the works in [10, 6, 1, 5, 18]; however in these works, fuzzy logic is essentially exploited to model similarities, but no attempt is done to build a global fuzzy retrieval condition and in implementing it (as done by our approach).

Because of the lack of space, we cannot discuss here specific examples or applications of our approach, neither to detail a comparison with similar approaches present in the literature: the interested reader can refer to [15] for such details.

We believe that the research on the strict relationships between CBR and fuzzy logic will eventually lead to the constructions of flexible reasoning systems, able to deal with problems of greater and greater complexity.

References


