# The POOR-MAD approach: Preferred Objects Over Rich, Multi-Attribute Data

(Discussion Paper)

Paolo Ciaccia[1], Davide Martinenghi[2] and Riccardo Torlone[3]

[1]*Dipartimento di Informatica - Scienza e Ingegneria, Università di Bologna, Viale Risorgimento, 2, 40136 Bologna, Italy*
[2]*Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio, 34/5, 20133 Milano, Italy*
[3]*Dipartimento di Ingegneria, Università Roma Tre, Via della Vasca Navale, 79, 00146 Roma, Italy*

**Abstract**

Preferences about objects of interest are often expressed at different levels of granularity, not always matching the level of detail of stored data. For instance, we prefer rock to pop music, yet scheduled concerts only cite the name of the performer, with no reference to the musical genre. In this paper, we address this common mismatch by leveraging the vast amounts of data organized in taxonomies (such as those found in electronic catalogs and classification systems). We present a model to represent preferences and state the desirable properties of preference propagation, such as the fact that more specific preferences always prevail over more generic ones. We then illustrate an approach for propagating preferences along taxonomies complying with the stated properties and show how the best objects can thereby be identified.

## 1. Introduction

The information available in digital form is growing so fast that the search for data of interest (for attending events, buying products, planning a trip, etc.) is becoming increasingly difficult over time. For this reason, there has recently been a huge effort to develop effective methods and tools able to automatically suggest to any individual the items that better match what he/she is looking for [1]. In this framework, the availability of preferences, explicitly expressed by the users or somehow automatically derived from their actions, has been always considered an important ingredient [2, 3]. Unfortunately, preferences and data do not always match perfectly, even when they refer to the same domain of interest. This is mainly due to the fact that, usually, preferences are expressed in generic terms whereas data is very specific, as shown in the next example.

*SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy*

paolo.ciaccia@unibo.it (P. Ciaccia); davide.martinenghi@polimi.it (D. Martinenghi); riccardo.torlone@uniroma3.it (R. Torlone)
http://www-db.deis.unibo.it/~pciaccia/ (P. Ciaccia); https://martinenghi.faculty.polimi.it (D. Martinenghi); http://www.dia.uniroma3.it/~torlone/ (R. Torlone)

**Example 1.** We are planning to reserve tickets for a series of concerts for which a general schedule is available, like the one in Figure 1. We prefer rock to pop concerts, yet we prefer a performance by Madonna to a rock concert. Due to work commitments, we also prefer concerts in August rather than in September. Furthermore, as for the concert venue, during autumn we prefer indoor places to stadiums. And, given two concerts by the same artist, we prefer to save money (say, if a concert costs less than 40$, then we prefer it to a concert by the same artist that costs more than 100$, whereas for intermediate prices other considerations are relevant). For the same reason, we would like to buy tickets only for (a subset of) the "best" available alternatives. ■

Concerts

| Artist | Day | Venue | Price ($) | |
|---|---|---|---|---|
| Bruce Springsteen | 10/05/2019 | Verona Arena | 70 | $t_a$ |
| Madonna | 24/06/2019 | Verona Arena | 35 | $t_b$ |
| Madonna | 21/07/2019 | Blue Note, Milan | 120 | $t_c$ |
| Eminem | 12/08/2019 | Unipol Arena, Bologna | 60 | $t_d$ |
| Rihanna | 10/10/2019 | Blue Note, Milan | 50 | $t_e$ |
| Bruce Springsteen | 30/10/2019 | Stadio Olimpico, Rome | 100 | $t_f$ |

**Figure 1:** A set of concerts.

The example highlights that: (i) preferences can be expressed at different levels of detail, even for the same "dimension" of the problem (e.g., seasons vs months for the time dimension), and (ii) in general, preferences do not match the level of detail of data. Moreover, preferences can be conflicting when changing the level of detail (rock is better than pop, yet Madonna, a pop singer, is preferred to rock artists). Finally, additional knowledge is needed to choose the best alternatives using preferences. For instance, we need to know that `Unipol Arena` is an indoor place, whereas `Verona Arena` is a Roman amphitheater (thus an outdoor place).

This problem can be tackled by leveraging the great availability of shared and public taxonomies, that is, collection of terms in a domain arranged hierarchically according to an inclusion relationship (e.g., product catalogs, book classifications, biological categorizations, etc.). For instance, the availability of a classification of music artists according to different musical genres would allow us to understand that a preference on rock artists *propagates* to Springsteen.

In this paper, which is an extended abstract of [4], we present a principled approach to the problem of finding the best objects stored in a data repository on the basis of a set of preferences that are defined at a level of detail that does not match that of the data. As a preliminary step, we adopt a data model for representing taxonomies of values in specific domains (e.g., time or location) and propose a preference model for tuples over a given set of taxonomies. We then identify the general properties of preference propagation in the taxonomies, in particular, the fact that more specific preferences prevail over more generic ones. Thus, in the example above, a Madonna concert takes precedence over a Springsteen concert even if, in general, we prefer rock to pop. We then illustrate an algorithm for propagating preferences along taxonomies, complying with the stated properties. Finally, we present a technique for selecting the best tuples according to the propagated preferences. This technique would select tuples $t_b$ and $t_d$ as the best alternatives among the tuples in Figure 1 given the preferences discussed in Example 1.

# 2. Preliminaries

## 2.1. Data Model

First of all, it is useful to remind that a *partial order* $\leq$ on a domain $V$ is a subset of $V \times V$, whose elements are denoted by $v_1 \leq v_2$, that is: reflexive ($v \leq v$ for all $v \in V$), antisymmetric (if $v_1 \leq v_2$ and $v_2 \leq v_1$ then $v_1 = v_2$), and transitive (if $v_1 \leq v_2$ and $v_2 \leq v_3$ then $v_1 \leq v_3$).

Our data model is a natural extension of the relational model in which the values in each domain can be arranged hierarchically according to a *taxonomy* [5, 6]. Each taxonomy is represented by a partial order $\leq_L$ on over a set $L$ of *levels*, each of which includes a set of values at a certain degree of granularity. For instance, a *time* taxonomy can be organized in levels such as day, month, season, and year, with day $\leq_L$ month $\leq_L$ year and day $\leq_L$ season. For each pair of levels $l_1 \leq_L l_2$ in a taxonomy, we then assume the existence of a function $\mu_{l_1}^{l_2}$, called *level mapping*, that maps each value in $l_1$ to a value in $l_2$. For instance, we have: $\mu_{\text{day}}^{\text{month}}(\texttt{23/07/2019}) = \texttt{07/2019}$. The level mappings induce a partial order $\leq_V$ on the values of a taxonomy where $v_1 \leq_V v_2$ if $l_1 \leq_L l_2$ and $\mu_{l_1}^{l_2}(v_1) = v_2$.

**Example 2.** Portions of the taxonomies relevant to our working example on concerts are shown in Figure 2. For the *location* of a concert we consider levels Venue, VenueType, and InOut. The values of this taxonomy are organized in the poset shown in Figure 3, where the level mappings are represented by arrows. ■

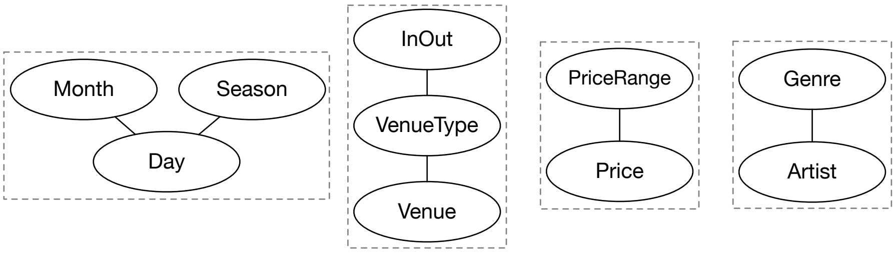

**Figure 2:** The taxonomies for our working example

The main constructs of the data model are the t-schema, the t-tuple, and the t-relation, which are natural extensions of the analogous notions in the relational model in which data domains can be taxonomies. For instance, a catalog of concerts in Italy can be represented by the t-relation shown in Figure 1 over the taxonomies in Figure 2 where we have used the levels as names of the attributes, assuming for simplicity that level names are unique. A partial order can than also be defined over t-schemas and t-tuples as follows: $S_1 \leq_S S_2$ if for each $l_i \in S_2$ there is an element $l_j \in S_1$ such that $l_j \leq_L l_i$ and $t_1 \leq_t t_2$ if: (i) $S_1 \leq_S S_2$, and (ii) for each $l_i \in S_2$ there is an element $l_j \in S_1$ such that $t_1[l_j] \leq_V t_2[l_i]$. Note that $S_2$ may have fewer

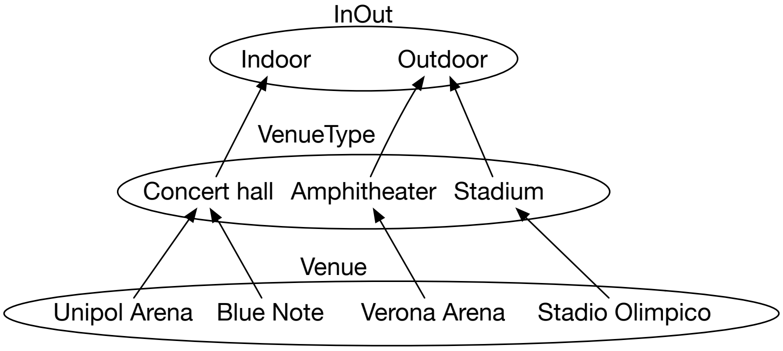

**Figure 3:** A taxonomy for concerts' locations

attributes than $S_1$. However, we assume without loss of generality that they have the same set of attributes, since we can add to $S_2$ the missing attributes at the top level of the taxonomy.

## 2.2. Preference Model

In our approach preferences are represented by a binary relation $\succeq$ over t-tuples as follows: given a collection of taxonomies $\{T_1, \ldots, T_k\}$ and a pair of t-tuples $t_1$ and $t_2$ over $\mathcal{T} = T_1 \times \ldots \times T_k$, if $t_1 \succeq t_2$ then we say that $t_1$ is *(weakly) preferable* to $t_2$. If $t_1 \succeq t_2$ and $t_2 \not\succeq t_1$ we say that $t_1$ is *strictly preferable* to $t_2$, denoted $t_1 \succ t_2$. Then, the "best" t-tuples in a t-relation $r$ according to the preference relation $\succeq$ can be selected by means of the *Best* operator $\beta_{\succeq}(r) = \{t \in r \mid \nexists t' \in r, t' \succ t\}$ [7].

We express preferences in a logic-based language, so that $t_1 \succeq t_2$ iff they satisfy the *preference formula* $F(t_1, t_2)$: $t_1 \succeq t_2 \Leftrightarrow F(t_1, t_2)$. In particular, we consider formulas in which only built-in predicates are present and quantifiers are omitted, also called *intrinsic preference formulas* (ipf) [2]. Furthermore, without loss of generality, we assume that $F$ is in Disjunctive Normal Form (DNF) and call each disjunct of $F$ a *preference clause*, i.e.: $F(t_1, t_2) = \bigvee_{i=1}^{n} P_i(t_1, t_2)$.

**Example 3.** The preferences informally stated in Example 1 can be expressed by the formula $F(t_1, t_2) = P_1(t_1, t_2) \vee \ldots \vee P_5(t_1, t_2)$, where:

$$\begin{aligned}
P_1(t_1, t_2) = \quad & (t_1[\text{Genre}] = \texttt{rock}) \wedge (t_2[\text{Genre}] = \texttt{pop}) \\
P_2(t_1, t_2) = \quad & (t_1[\text{Artist}] = \texttt{Madonna}) \wedge (t_2[\text{Genre}] = \texttt{rock}) \\
P_3(t_1, t_2) = \quad & (t_1[\text{Artist}] = t_2[\text{Artist}]) \wedge \\
& (t_1[\text{PriceRange}] = \texttt{cheap}) \wedge (t_2[\text{PriceRange}] = \texttt{expensive}) \\
P_4(t_1, t_2) = \quad & (t_1[\text{Season}] = \texttt{autumn}) \wedge (t_2[\text{Season}] = \texttt{autumn}) \wedge \\
& (t_1[\text{InOut}] = \texttt{indoor}) \wedge (t_2[\text{VenueType}] = \texttt{stadium}) \\
P_5(t_1, t_2) = \quad & (t_1[\text{Month}] = \texttt{august}) \wedge (t_2[\text{Month}] = \texttt{september})
\end{aligned}$$

■

# 3. Propagation of Preferences

The initial preference relation $\succeq$ completely ignores the structure of the poset on t-tuples $\leq_t$, thus it treats $\mathcal{T}$ as if it were a "flat" domain. This is because $\succeq$ includes all and only those preferences $t_1 \succeq t_2$ such that the attribute values of $t_1$ and $t_2$ satisfy the preference formula, yet it does not take into account the taxonomies. The key observation justifying the *downward propagation* of preferences, is that, given a target t-schema $S$, by exploiting the hierarchical organization of $\mathcal{T}$ it is possible to extend $\succeq$ with more preferences that involve t-tuples with t-schemas $S_i$, with $S \leq_S S_i$.

Let $\succeq_\mathsf{D}$ denote the binary relation obtained by propagating, in some way to be described, the preferences in $\succeq$. The basic idea underlying propagation is that, if we have t-tuples $t'_1$ and $t'_2$ such that $t'_1 \succeq t'_2$, then this preference can be downward propagated to all t-tuples $t_1$ and $t_2$ such that $t_1 \leq_t t'_1$ and $t_2 \leq_t t'_2$. For instance, consider the t-schema in Figure 1 and the preference clause $P_1(t_1, t_2) = t_1[\mathsf{Genre}] = \mathsf{Rock}) \wedge (t_2[\mathsf{Genre}] = \mathsf{Pop})$. From $P_1$ we can obtain through propagation the preferences $t_a \succeq_\mathsf{D} t_b$ and $t_a \succeq_\mathsf{D} t_c$, since Bruce Springsteen is a rock artist whereas Madonna is a pop singer.

In terms of logical formulas, downward propagation with respect to the target t-schema $S$ can be easily obtained by exploiting the level mappings. For instance, consider clause $P_5(t_1, t_2)$ in Example 3. This can be rewritten as:

$$(\mu_{\mathsf{Day}}^{\mathsf{Month}}(t_1[\mathsf{Day}]) = \texttt{august}) \wedge (\mu_{\mathsf{Day}}^{\mathsf{Month}}(t_2[\mathsf{Day}]) = \texttt{september})$$

so that it is applicable to values at the Day level. In order to simplify the notation, in the following level mappings are understood and we use $=_\mathsf{D}$ in place of $=$ to allow comparisons between values at different levels (similarly for other comparison operators, eg., $\leq$ would become $\leq_D$). Thus, the above clause can be more conveniently written as:

$$t_1[\mathsf{Day}] =_\mathsf{D} \texttt{august} \wedge t_2[\mathsf{Day}] =_\mathsf{D} \texttt{september}.$$

Given an input formula $F$, we denote by $F_\mathsf{D}$ the formula rewritten by means of level mappings.

## 3.1. Transitively Closing the Formula

Downward propagation does not guarantee that $\succeq_\mathsf{D}$ is a transitive relation, even because the very input relation $\succeq$ is not necessarily transitive. Although one may be tempted to circumvent this problem by adopting an algorithm for non-transitive preferences [8], algorithms of this type can discard a sub-optimal t-tuple $t$ only if the t-relation $r$ contains a t-tuple $t'$ that is *directly* (rather than transitively) better than $t$.

Our approach is to transitively closing formula $F_\mathsf{D}$ with respect to the t-tuple domain $\mathcal{T}$, as described in [2] for the base case of flat domains.[1] The presence of taxonomies requires to extend the basic scheme adopted in [2], as the following example illustrates.

[1] Notice that the transitive closure of an ipf is finite and still an ipf [2].

**Example 4.** Given the formula $F(t_1, t_2) = P_5(t_1, t_2) \vee P_4(t_1, t_2)$, where $P_4$ and $P_5$ are as in Example 3, by means of downward propagation we obtain the formula:

$$\begin{aligned} F_{\mathsf{D}}(t_1, t_2) = \;& (t_1[\mathsf{Day}] =_{\mathsf{D}} \texttt{august} \wedge t_2[\mathsf{Day}] =_{\mathsf{D}} \texttt{september}) \vee \\ & (t_1[\mathsf{Day}] =_{\mathsf{D}} \texttt{autumn} \wedge t_2[\mathsf{Day}] =_{\mathsf{D}} \texttt{autumn} \wedge \\ & t_1[\mathsf{Venue}] =_{\mathsf{D}} \texttt{indoor} \wedge t_2[\mathsf{Venue}] =_{\mathsf{D}} \texttt{stadium}). \end{aligned}$$

Now, since some `september` days are in `autumn`, it is sound to add to the transitive closure of $F_{\mathsf{D}}$, a formula that will be denoted as $F_{\mathsf{DT}}$, the clause:

$$t_1[\mathsf{Day}] =_{\mathsf{D}} \texttt{august} \wedge t_2[\mathsf{Day}] =_{\mathsf{D}} \texttt{autumn} \wedge t_2[\mathsf{Venue}] =_{\mathsf{D}} \texttt{stadium}.$$

■

We observe that $F_{\mathsf{D}}$ uses level mappings in order to understand when preferences expressed at different levels can be transitively combined. Indeed, this is the only additional complexity with respect to the procedure given in [2]. In particular, similarly to what done in Example 4, we have to compare values at different levels in a taxonomy (e.g., `september` and `autumn`) and determine if they have a non-empty intersection (so that the transitive step can be applied).

## 3.2. Computing the Result with Specific Preferences

The propagation scheme described in the previous section is unable to deal with conflicting preferences, one of which is *more specific* than the other. In our working example, we have a generic preference for rock concerts over pop concerts (clause $P_1$), yet we also have a more specific preference stating that a performance by Madonna takes precedence over any rock concerts (clause $P_2$). Thus, according to $P_1$ we would have, among others, $t_a \succeq_{\mathsf{DT}} t_b$, whereas $P_2$ would yield $t_b \succeq_{\mathsf{DT}} t_a$, thus making $t_a$ and $t_b$ indifferent. We argue that giving the same importance to both preferences contradicts the intuition, as the more specific preference should take precedence over the more generic one.

We now detail how, given a formula $F_{\mathsf{DT}}$, we can effectively solve conflicts by maintaining only more specific preferences.

If $F_{\mathsf{DT}}(t_1, t_2)$ holds, we look at the clause $P_i$ that evaluates to **true**[2] and, for each involved attribute, we consider the original level it has for both $t_1$ and $t_2$ (remind that $F_{\mathsf{DT}}$ has been obtained by applying level mappings). If an attribute does not appear in $P_i$ we set its level to the top level of its taxonomy (henceforth simply indicated by $\top$).

**Example 5.** Consider t-tuples $t_a$ and $t_b$ in Figure 1 and clause $P_2(t_1, t_2) = (t_1[\mathsf{Artist}] = \texttt{Madonna}) \wedge (t_2[\mathsf{Genre}] = \texttt{rock})$. Note that $P_2(t_b, t_a)$ holds and the original levels when evaluating $P_2$ for $t_a$ are $(\mathsf{Genre}, \top, \top, \top)$, while those for $t_b$ are $(\mathsf{Artist}, \top, \top, \top)$. ■

Overall, this leads to *a pair of t-schemas*, $\mathsf{spp}(t_1, t_2) = \langle \mathsf{sig}_{1,2}(t_1), \mathsf{sig}_{1,2}(t_2) \rangle$. Notice that $\mathsf{sig}_{1,2}(t_1)$ is the t-schema for $t_1$ when we test if $t_1$ is preferred to $t_2$, which, in general, is different from $\mathsf{sig}_{2,1}$, i.e., the t-schema for $t_1$ when we test if $t_2$ is preferred to $t_1$. This motivates the use of subscripts.

[2] Generalization to the case in which more than one clause is **true** is immediate.

**Algorithm 1:** Computing the best t-tuples in $r$.

```
Input: t-relation $r$ with t-schema $S = \{A_1 : l_1, \ldots, A_k : l_k\}$, formula $F_{\mathsf{DT}}$.
Output: $\beta_{\succ_{\mathsf{DT}}}$.
let $Best := \emptyset$
for each $t \in r$
  let $Opt :=$ true
  for each $t' \in Best$
    cases
      $F_{\mathsf{DT}}(t, t') \wedge (F_{\mathsf{DT}}(t', t) \wedge t = \textsc{MoreSpecificPref}(t, t') \vee \neg F_{\mathsf{DT}}(t', t))$ :
                                      $Best := Best \setminus \{t'\}$
      $F_{\mathsf{DT}}(t', t) \wedge (F_{\mathsf{DT}}(t, t') \wedge t' = \textsc{MoreSpecificPref}(t, t') \vee \neg F_{\mathsf{DT}}(t, t'))$ :
                                      let $Opt :=$ false; break
  if $Opt$ then $Best := Best \cup \{t\}$
return $Best$
```

When also $F_{\mathsf{DT}}(t_2, t_1)$ holds, i.e., we have a conflict, we compare $\mathsf{spp}(t_1, t_2)$ and $\mathsf{spp}(t_2, t_1) = \langle \mathsf{sig}_{2,1}(t_2), \mathsf{sig}_{2,1}(t_1) \rangle$ and conclude that the first preference is more specific than the second iff $\mathsf{sig}_{1,2}(t_1) \leq_S \mathsf{sig}_{2,1}(t_1)$ and $\mathsf{sig}_{1,2}(t_2) \leq_S \mathsf{sig}_{2,1}(t_2)$, with at least one t-schema being strictly more specific.

**Example 6.** Assume we are comparing t-tuples $t_a$ and $t_b$ in Figure 1. From clause $P_1$ in Example 3 ($P_1(t_1, t_2) = (t_1[\mathsf{Genre}] = \mathtt{rock}) \wedge (t_2[\mathsf{Genre}] = \mathtt{pop})$) we derive that $t_a \succeq_{\mathsf{DT}} t_b$, whereas $t_b \succeq_{\mathsf{DT}} t_a$ follows from clause $P_2$. For the first preference we have $\mathsf{spp}(t_a, t_b) = \langle(\mathsf{Genre}, \top, \top, \top), (\mathsf{Genre}, \top, \top, \top)\rangle$, whereas for the second we have $\mathsf{spp}(t_b, t_a) = \langle(\mathsf{Artist}, \top, \top, \top), (\mathsf{Genre}, \top, \top, \top)\rangle$. Although for $t_a$ the two t-schemas are the same, for $t_b$ Artist is strictly more specific than Genre, thus $t_b$ is strictly preferred to $t_a$. ■

In order to compute the best results according to the (transitively closed) rewritten formula $F_{\mathsf{DT}}$ we can use any algorithm developed for returning the best objects in a strict partial order, such as those in [9, 7], by suitably adapting it to work in our scenario. Algorithm 1 is such an adaptation of the well-known BNL algorithm [9]. In the algorithm the "specificity test" is performed by the procedure $\textsc{MoreSpecificPref}(t, t')$, which returns $t$ if the preference $t \succeq_{\mathsf{DT}} t'$ is more specific than $t' \succeq_{\mathsf{DT}} t$, $t'$ in the opposite case, and **nil** otherwise.

## 4. Conclusions

In this paper we have studied preference propagation along several taxonomies, when the levels at which preferences are stated and that of the stored data differ. The preference model we have proposed is able to deal with conflicting preferences in an effective way, thus propagating only the most specific preferences.

The specificity principle we use in this paper was also considered in [10, 11], although on a preference model using strict rather than weak preferences and in a different scenario regarding preferences combined across different *contexts* (not preference formulas): if $a \succ b$ holds in context $c$, and $b \succ a$ in context $c'$, then $a \succ b$ prevails if $c$ is more specific than $c'$. The problem of dealing with preferences defined on different schemas, which is the main focus of the present paper, was not addressed at all in [10, 11].

The interplay between specificity and transitivity is studied in depth in [12].

Propagation of preferences in OLAP systems is considered in [13], where an algebraic language is adopted. Propagation occurs along hierarchies of levels, however no issue concerning combination of conflicting preferences is considered. Unlike most works studying the problem of managing *qualitative* preference queries on databases [3], in which the preference relation is a strict partial order $\succ$, in this paper we have considered “weak" preferences $\succeq$. This choice originates from the observation that, while propagating preferences between different t-schemas, transitivity cannot be guaranteed and a transitive closure is needed. However, enforcing transitivity might lead to cycles, which are harmless in our model but cannot occur in strict partial orders.

Future work includes the study of efficient methods for computing the transitive closure of the preference formula, the development of ad hoc algorithms for determining the best objects that scale over very large datasets, and an experimental evaluation on real-world scenarios.

## References

[1] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: Recommender Systems Handbook, Springer, 2011, pp. 1–35.

[2] J. Chomicki, Preference formulas in relational queries, TODS 28 (2003) 427–466.

[3] K. Stefanidis, G. Koutrika, E. Pitoura, A survey on representation, composition and application of preferences in database systems, TODS 36 (2011) 19:1–19:45.

[4] P. Ciaccia, D. Martinenghi, R. Torlone, Finding preferred objects with taxonomies, in: ER, 2019, pp. 397–411.

[5] D. Martinenghi, R. Torlone, Querying databases with taxonomies, in: ER, 2010, pp. 377–390.

[6] D. Martinenghi, R. Torlone, Taxonomy-based relaxation of query answering in relational databases, VLDB J. 23 (2014) 747–769.

[7] R. Torlone, P. Ciaccia, Which are my preferred items?, in: RPEC, 2002, pp. 217–225.

[8] C. Y. Chan, H. V. Jagadish, K. Tan, A. K. H. Tung, Z. Zhang, Finding k-dominant skylines in high dimensional space, in: SIGMOD, 2006, pp. 503–514.

[9] S. Börzsönyi, D. Kossmann, K. Stocker, The skyline operator, in: ICDE, 2001, pp. 421–430.

[10] P. Ciaccia, D. Martinenghi, R. Torlone, Foundations of context-aware preference propagation, J. ACM 67 (2020) 4:1–4:43.

[11] P. Ciaccia, R. Torlone, Modeling the propagation of user preferences, in: ER, 2011, pp. 304–317. *Best paper award.*

[12] P. Ciaccia, D. Martinenghi, R. Torlone, Preference queries over taxonomic domains, Proceedings of the VLDB Endowment (2021) to appear.

[13] M. Golfarelli, S. Rizzi, P. Biondi, myOLAP: An approach to express and evaluate OLAP preferences, TKDE 23 (2011) 1050–1064.