

Modeling the Propagation of User Preferences

Paolo Ciaccia¹ and Riccardo Torlone²

¹ Dip. di Elettronica, Informatica e Sistemistica, Università di Bologna, Italy
`paolo.ciaccia@unibo.it`

² Dip. di Informatica e Automazione, Università Roma Tre, Italy
`torlone@dia.uniroma3.it`

Abstract. User preferences are a fundamental ingredient of personalized database applications, in particular those in which the user context plays a key role. Given a set of preferences defined in different contexts, in this paper we study the problem of deriving the preferences that hold in one of them, that is, how preferences *propagate* through contexts. For the sake of generality, we work with an abstract context model, which only requires that the contexts form a poset. We first formalize the basic properties of the propagation process: *specificity*, stating that more specific contexts prevail on less specific ones, and *fairness*, stating that this behavior does not hold for incomparable contexts. We then introduce an algebraic model for preference propagation that relies on two well-known operators for combining preferences: Pareto and Prioritized composition. We study three alternative propagation methods and precisely characterize them in terms of the fairness and specificity properties.

1 Introduction

The information available in digital form is growing so fast that today the availability of methods for automatically filtering the accessible data according to the real needs of the users has become a compelling need. In this framework, context awareness [5] and user preferences [6] have emerged as viable solutions to this problem. The former refers to the ability of selecting data according to features of the environment in which the application is used, such as the location, the time, and the device. The latter refers to the ability of evaluating the relevance of data for a given user on the basis of a set of preferences settled on such data.

In this paper we consider both aspects together and study the problem of selecting the most relevant data in a situation in which preferences are defined in different contexts and queries are posed in one of them. The scenario we refer to is illustrated in the following example.

Example 1. Assume that we have fixed some contextual preferences for food such as “In Italy, a dish of pasta is preferable to one of beef, but if you are in Naples you should taste the world-famous pizza instead of pasta. In summer, however, a fresh salad is preferable to a hot dish of pasta”. Assume now that it is summer, we are in Naples and we would like to have some suggestion for food. All of such preferences should be taken into account since they refer to contexts that are

more general than the current one. However, it is evident that the preferences defined in “Naples” and “Italy in summer” should take precedence on those in the more generic context “Italy”. Moreover, the preference in “Naples” should not take precedence on the preference in “Italy in summer”, and vice versa, since, in general, one does not apply to the other. It turns out that, in the current context, pizza and salad are both the best alternatives among the mentioned foods and should be returned first by a data filtering system since, on the basis of the given preferences, no other food is preferable to them.

As shown in the example, a generalization hierarchy can be usually defined over contexts and preferences defined in different contexts *propagate* along this hierarchy. Thus the problem can be rephrased as the investigation of preference propagation in a hierarchy of contexts and its impact on database querying. Recently, this issue has been studied extensively [1,11,14,15,16,18] but always resorting to pragmatic, operational approaches. Conversely, we intend to tackle the problem in a principled way with the goal of providing a solid basis to the issue of context-aware preferences in database applications.

With this goal in mind, we consider a very general framework in which the only requirement is that the contexts form a *poset*, that is, a set with a partial order relationship on its elements. Actually, in order to provide concrete examples we introduce a formalism (the CT model) for the representation of contexts, but our results apply to the general notion of context poset. Moreover, we express user preferences in terms of a *partial* order relation over the tuples of interest [6,8], a general approach that includes the case in which preferences are expressed by associating a numerical score with tuples [2,10].

We then start by formalizing the basic properties of the propagation process, which are also implicitly at the basis of earlier approaches to the problem [14,15,16] and correspond to the observations made in Example 1: (i) preferences over a pair of incomparable contexts c_1 and c_2 do not take precedence on each other (*fairness*), and (ii) preferences over a context c_1 take precedence on the preferences over a context c_2 when c_1 is more specific than c_2 (*specificity*).

Building on these notions, we introduce an algebraic model for expressing preference propagation based on two binary operators: (i) \oplus , which combines preferences defined in two unordered contexts, and (ii) \otimes , which combines preferences in two ordered contexts. Interestingly, it turns out that these operators can be captured by two popular methods for combining preferences: Pareto and Prioritized composition [6,8]. We then adopt their semantics for \oplus and \otimes and call a composition thereof a PC (Preference Composition) expression.

Example 2. An example of PC-expression computing the propagation of preferences to the context “Naples in summer” discussed in Example 1 is the following:

$$\succ_{\text{Naples,summer}} \otimes ((\succ_{\text{Naples}} \oplus \succ_{\text{Italy,summer}}) \otimes \succ_{\text{Italy}})$$

where \succ_c denotes a set of preferences defined in the context c and, e.g., $\succ_{\text{Italy}} = \{\text{pasta} \succ \text{beef}\}$. In this expression the preferences in “Naples” and those in “Italy in summer” are combined with \oplus , since the corresponding contexts are

incomparable. The result is combined with the preferences in “Italy” using \oplus , since this context is more general than both “Naples” and “Italy in summer”. Finally, the result is combined with the preferences for “Naples in summer” using \otimes , since it is the most specific context.

We identify a “natural” form of PC-expression and prove that it is unable to enforce specificity. This leads us to introduce two alternative approaches to preference propagation. It turns out that the latter is indeed well-behaved since it satisfies all the desirable properties for preference propagation.

In sum, our main contributions are: (i) the identification and formalization of the desirable properties of preference propagation in a poset of contexts; (ii) the definition of an algebra for preference propagation based on Prioritized and Pareto composition; and (iii) the formal analysis of three propagation methods. To our knowledge, these are the first results that can provide a theoretical foundation to the management of contextual preferences in database systems.

The rest of the paper is organized as follows. In Section 2, we introduce the notion of context poset and present a specific context model that is used in the examples. In Section 3 we introduce the algebraic model for combining preferences and, in Section 4, we investigate different methods for computing preference propagation. In Section 5 we compare our work with the related literature and finally, in Section 6, we draw some conclusions.

2 Contexts in Databases

2.1 A General Notion of Context

Our aim is to investigate the problem of contextual preference queries independently of the specific formalism used to represent contexts (and preferences as well). We only focus on a fundamental characteristic of context models: the ability to represent contexts at different levels of detail [4]. We will therefore rely on the general notion of context that follows.

We recall that a *partial order* \leq_V on a domain of values V is a binary relation on V that is reflexive ($v \leq_V v$ for all $v \in V$), antisymmetric (if $v_1 \leq_V v_2$ and $v_2 \leq_V v_1$ then $v_1 = v_2$), and transitive (if $v_1 \leq_V v_2$ and $v_2 \leq_V v_3$ then $v_1 \leq_V v_3$). A partially ordered set, or *poset*, is a set S on which a partial order \leq_S is defined.

Definition 1 (Context). A context c is an element of a poset C , called context poset. If $c_1 \leq_C c_2$ and $c_1 \neq c_2$, written $c_1 <_C c_2$, we say that c_1 is more specific than c_2 and that c_2 is more generic than c_1 .

2.2 The CT Model

The CT (ConTextual) model is a possible refinement of the deliberately general notion of context poset introduced above. In CT a context is represented by means of a finite set of *contextual dimensions*, which represent aspects that may influence the relevance of data, such as time and location. Each dimension is described by means of a set of *levels* representing the dimension at different degrees of granularity.

Definition 2 (Contextual dimension). A (CT) contextual dimension d is composed of: (i) a finite poset L of levels; each level l in L is associated with a set of values $M(l)$, called the members of level l ; and (ii) a family CM of containment mappings $\text{CMAP}_{l_1}^{l_2} : M(l_1) \rightarrow M(l_2)$ for each $l_1 \leq_L l_2$.

For instance, day, week, month, and year are possible levels for the time dimension. A possible member of the day level is 23/07/2011, which is mapped by the containment mappings to the member 07/2011 of the level month.

A partial order \leq_M can also be defined on the members M of a dimension d : given a pair of members m_1 and m_2 of levels l_1 and l_2 of d , respectively, we have that $m_1 \leq_M m_2$ if $\text{CMAP}_{l_1}^{l_2}(m_1) = m_2$. We are now ready to introduce the notion of context poset in the CT model.

Definition 3 (Context poset in the CT model). Let D be a finite set of dimensions. A (CT) context c over D is a tuple over l_1, \dots, l_n , where each l_i is a level of a dimension in D . A CT context poset C over D is a set of contexts c_j over some $D_j \subseteq D$, such that $c_1 \leq_C c_2$ if, for each dimension $d \in D_2$: (i) c_1 is defined on d , and (ii) $c_1[d] \leq_M c_2[d]$, where $c[d]$ denotes the member of d occurring in a context c .

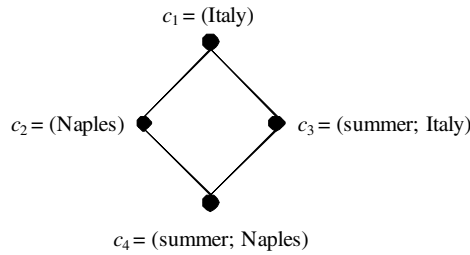


Fig. 1. A context poset in the CT model

Example 3. A simple example of CT context poset, which refers to the scenario discussed in Example 1, is shown in Figure 1.¹ In this example, c_1 is a context over the dimension Location at the level Country, whereas c_2 is a context over the same dimension at level City. Similarly, c_3 and c_4 are contexts over dimensions Time and Location at the levels {Season, Country} and {Season, City}, respectively. Then, we have for instance that $c_4 \leq_C c_3$ since summer \leq_M summer and Naples \leq_M Italy.

3 Preferences in Contexts

3.1 Qualitative Preferences

According to a general and widely accepted approach [6,8], preferences over tuples of a relation scheme $R(A_1, \dots, A_k)$ are expressed by a binary relation

¹ We represent the poset with its *Hasse diagram*, in which the edges represent the partial order (transitively reduced), and x is drawn lower than y if $x \leq y$.

over the *tuple domain* of R , that is, over the set $\prod_{i=1}^k D_i$, where D_i is the value domain of attribute A_i . We recall that a *strict* partial order on a set S is an asymmetric (we never have both $s_1 <_S s_2$ and $s_2 <_S s_1$) and transitive binary relation on S .

Definition 4 (Preference relation). A preference relation \succ over a scheme $R(X)$ is a strict partial order on the tuple domain of R . Given a pair of tuples t_1 and t_2 in the tuple domain of R , if $t_1 \succ t_2$ then t_1 is preferable to t_2 . If neither $t_1 \succ t_2$ nor $t_2 \succ t_1$, then t_1 and t_2 are indifferent, denoted by $t_1 \sim t_2$.

A refinement of the indifference relation \sim associated to a preference relation \succ allows some indifferent tuples to be also *equivalent*, which, as we will see, is a key property for the composition of preference relations [9].

Definition 5 (Equivalent tuples). Given a preference relation \succ , t_1 and t_2 are equivalent under \succ , written $t_1 \approx t_2$, if $t_1 \sim t_2$ and, for all t in the tuple domain such that $t_1 \succ t$ ($t \succ t_1$), it is $t_2 \succ t$ ($t \succ t_2$), and vice versa. If $t_1 \sim t_2$, but $t_1 \not\approx t_2$, we say that t_1 and t_2 are incomparable.

Example 4. Let us consider the relation over the scheme $R(\text{Food}, \text{Calories}, \text{Fat})$ made of the following tuples:

$$t_1 = (\text{pasta}, 221, 1.3) \quad t_2 = (\text{beef}, 63, 2.5) \quad t_3 = (\text{salad}, 15, 0.1) \quad t_4 = (\text{pizza}, 160, 3.2)$$

A possible set of preferences over this scheme is $t_1 \succ t_2$ and $t_3 \succ t_2$. In words, pasta and salad are both preferable to beef. It follows that pasta and salad are equivalent, whereas pizza is incomparable with all other foods.

The best tuples in a relation r over the scheme $R(X)$ according to the preferences \succ can be selected by the *Best* operator [17]: $\beta_{\succ}(r) = \{t \in r \mid \nexists t' \in r, t' \succ t\}$.² The restriction of Definition 4 to strict partial orders guarantees that, for any $r \neq \emptyset$, $\beta_{\succ}(r)$ is never empty [6]. A *preference query* is any expression of the relational algebra augmented with the Best operator.

3.2 Contextual Preferences and Their Propagation

Throughout the paper we consider a context poset C and a relation scheme $R(X)$, and denote by BP a function that associates with each context $c \in C$ a preference relation $\text{BP}(c) = \succ_c$ over $R(X)$.³ If no preferences have been defined for a context c in C , then $\text{BP}(c) = \emptyset$. We call \succ_c the *base* preferences in c and BP a *preference configuration* over C . Since, as we have seen, preferences *propagate* along the poset C , we call *complete* preferences in c , denoted by \succ_c^+ , the result of combining \succ_c with the base preferences defined in the other contexts in C , and denote by CP the function that associates with each context $c \in C$ the complete preferences $\text{CP}(c) = \succ_c^+$ over $R(X)$. We call CP the *propagation* (function) of a preference configuration BP over a context C .

² This operator is also called *winnow* [6] and *preference selection* [8].

³ We assume that BP is given by the user or somehow derived from the application.

Let us now try to capture the basic ideas underlying earlier, practical approaches on preference propagation [14,15,16]. The first issue is the *scope* of propagation. It is apparent that, for each context c , all and only the base preferences in the contexts c' such that $c \leq_C c'$ (denoted by $C[c]$) are needed to determine \succ_c^+ . This is made precise by the following definition.

Definition 6 (Propagation scope). *The propagation scope of a context $c \in C$ is $C[c]$. For each c' in the scope of c , if $t_1 \succ_{c'} t_2$ and $t_1 \approx_{c'} t_2$ for each $c'' \in C[c]$ different from c' , then $t_1 \succ_c^+ t_2$.*

The second part of the definition specifies that a preference between two tuples in a context c' , belonging to the scope of c , has to be propagated in c if such tuples are equivalent in all the other contexts in the scope of c .

As discussed in Example 1, two further basic properties should be satisfied by preference propagation. Specifically, given a context c : (1) for each pair of unordered contexts c_1 and c_2 in the scope of c , the base preferences in c_1 and c_2 should not take precedence on each other in determining \succ_c^+ ; in this case we say that the propagation is *fair*; (2) for each pair of ordered contexts $c_1 <_C c_2$ in the scope of c , the base preferences in c_1 should take precedence on those in c_2 in determining \succ_c^+ ; in this case we say that the propagation is *specific*. A precise characterization of the above principles can be given as follows.

Definition 7 (Fairness). *A propagation CP is fair if, for each context c in C , each pair of unordered contexts c_1 and c_2 in $C[c]$, and each preference configuration BP over C such that: (i) $t_1 \succ_{c_1} t_2$, (ii) $t_2 \succ_{c_2} t_1$, (iii) $t_1 \approx_{c_i} t_2$ for each $c \leq_C c_i <_C c_1$, and (iv) $t_1 \approx_{c_j} t_2$ for each $c \leq_C c_j <_C c_2$, it is: $t_1 \not\succeq_c^+ t_2$ and $t_2 \not\succeq_c^+ t_1$.*

Definition 8 (Specificity). *A propagation CP is specific if, for each context c in C , each pair of ordered contexts $c_1 <_C c_2$ in $C[c]$, and at least one preference configuration BP over C such that: (i) $t_1 \succ_{c_1} t_2$, (ii) $t_2 \succ_{c_2} t_1$, (iii) $t_1 \approx_{c_i} t_2$ for each $c \leq_C c_i <_C c_1$, it is: $t_1 \succ_c^+ t_2$.*

Basically, Definition 7 asserts that if c_1 and c_2 disagree on how to order t_1 and t_2 while such tuples are equivalent in all more specific contexts, then t_1 and t_2 are not ordered in \succ_c^+ . Conversely, Definition 8 says that if t_1 is preferred to t_2 in c_1 and such tuples are equivalent in all more specific contexts, then this preference might propagate to context c (and it actually propagates if no conflict arises with other contexts in BP incomparable with c_1).

3.3 PC-Expressions

The properties of specificity and fairness suggest that the complete preferences can be computed by means of an expression involving two basic binary operators that, given two base preference relations \succ_{c_1} and \succ_{c_2} , return a new preference relation: one applies when c_1 and c_2 are unordered, the other one when $c_1 <_C c_2$. Clearly, the former is commutative whereas this is not the case for the latter. Both operators are associative, since their composition should not depend on

the order in which preferences are considered. Also, they are both idempotent since the combination of the same preferences should not have any effect. Finally, they both have the empty set of preferences, \emptyset , as identity element since contexts without base preferences should not influence the result.

Incidentally, there are two popular ways to combine preference relations that meet all the requirements above: Pareto and Prioritized composition [6,8].

Definition 9 (Pareto and Prioritized composition). *Let \succ_1 and \succ_2 be two preference relations over a scheme $R(X)$. The Prioritized composition of \succ_1 and \succ_2 , written $\succ_1 \otimes \succ_2$, is defined as:*

$$t_1 \succ_1 \otimes \succ_2 t_2 \Leftrightarrow (t_1 \succ_1 t_2) \vee (t_1 \succ_2 t_2 \wedge t_1 \approx_1 t_2)$$

and their Pareto composition, written $\succ_1 \oplus \succ_2$, is:

$$t_1 \succ_1 \oplus \succ_2 t_2 \Leftrightarrow (t_1 \succ_1 t_2 \wedge t_1 \succ_2 t_2) \vee (t_1 \succ_1 t_2 \wedge t_1 \approx_2 t_2) \vee (t_1 \approx_1 t_2 \wedge t_1 \succ_2 t_2)$$

Intuitively, Prioritized composition gives precedence to preferences in \succ_1 , while preferences in \succ_2 are used only if two tuples are equivalent according to \succ_1 . Conversely, Pareto considers the two preference relations equally important.

Example 5. Given the two preference relations

$$\succ_1 = \{t_1 \succ_1 t_2, t_1 \succ_1 t_3\} \quad \succ_2 = \{t_2 \succ_2 t_1, t_2 \succ_2 t_3\}$$

it is $t_2 \approx_1 t_3$ and $t_1 \approx_2 t_3$. Then, the Prioritized composition $\succ_1 \otimes \succ_2$ yields the preferences $t_1 \succ t_2$, $t_1 \succ t_3$, and $t_2 \succ t_3$, whereas the Pareto composition $\succ_1 \oplus \succ_2$ leads to have $t_1 \succ t_3$ and $t_2 \succ t_3$.

Actually, both Prioritized and Pareto composition preserve strict partial orders [9], whereas this is not guaranteed by replacing in their definition \approx with \sim [6]. It is known that \oplus is commutative and associative and that \otimes is associative [9] (but obviously not commutative). It is also evident that both operators are idempotent (that is, $\succ \oplus \succ = \succ$ and $\succ \otimes \succ = \succ$) and have \emptyset as the identity.

We are now ready to introduce the main tool for expressing preference propagation in our framework.

Definition 10 (PC-expression). *A preference composition expression, or PC-expression, is any expression E of the form: $E ::= \succ_c \mid E \oplus E \mid E \otimes E \mid (E)$.*

Because of Definition 6, any PC-expression computing the complete preferences in a context c must include all and only the base preferences in contexts in $C[c]$.

Example 6. Consider the context poset in Figure 1. The complete preferences for context c_4 could be expressed by the following PC-expression:

$$\succ_{c_4} \otimes ((\succ_{c_2} \otimes \succ_{c_1}) \oplus (\succ_{c_3} \otimes \succ_{c_1}))$$

Both \succ_{c_2} and \succ_{c_3} are first combined with \succ_{c_1} using the \otimes operator, since the corresponding contexts are ordered. These sub-expressions are then combined with the \oplus operator, since c_2 and c_3 are unordered. Finally, \succ_{c_4} is added using the \otimes operator, since c_4 is the most specific context.

4 Computing the Propagation of Preferences

4.1 The Complete-Cover Propagation

The first way of using a PC-expression for computing preference propagation, which we call \mathcal{CC} (Complete Cover), is based on an intuitive argument: the complete preferences in a context c can be obtained recursively by composing the base preferences in c , \succ_c , with the complete preferences that hold in the contexts that, in the context poset C , cover c . We remind that c_i covers c if $c <_C c_i$ and there is no other context c_j such that $c <_C c_j <_C c_i$. The covering of c in the poset C , denoted $\text{COV}_C(c)$, is the set of contexts in C that cover c .

Definition 11. Let c be a context in C with covering $\text{COV}_C(c) = \{c_1, \dots, c_k\}$. The complete preferences in c under the \mathcal{CC} propagation are computed as:

$$\succ_c^{+\mathcal{CC}} = \begin{cases} \succ_c \otimes (\succ_{c_1}^{+\mathcal{CC}} \oplus \succ_{c_2}^{+\mathcal{CC}} \oplus \dots \oplus \succ_{c_k}^{+\mathcal{CC}}) & \text{if } \text{COV}_C(c) \neq \emptyset \\ \succ_c & \text{if } \text{COV}_C(c) = \emptyset \end{cases} \quad (1)$$

From Equation 1 we can derive a “canonical” PC-expression. For this, the following preliminary result is fundamental.

Lemma 1. Prioritized composition left-distributes (but not right-distributes) over Pareto composition, that is, for any preference relations $\succ_1, \succ_2, \succ_3$ it is:

$$\succ_1 \otimes (\succ_2 \oplus \succ_3) = (\succ_1 \otimes \succ_2) \oplus (\succ_1 \otimes \succ_3)$$

For the following result we remind that a chain $H = \langle c_1, \dots, c_k \rangle$ of poset C is a sequence of contexts such that $c_1 <_C c_2 <_C \dots <_C c_k$, and that H is maximal if it is not included into another chain.

Corollary 1. Let c be a context in C and $\text{COV}_C(c) = \{c_1, \dots, c_k\}$ be the covering of c in C . The complete preferences in c under the \mathcal{CC} propagation can also be computed by the canonical PC-expression:

$$\succ_c^{+\mathcal{CC}} = \otimes(H_1) \oplus \otimes(H_2) \oplus \dots \oplus \otimes(H_l) \quad (2)$$

where $\{H_1, H_2, \dots, H_l\}$ are all the maximal chains in $C[c]$ (the set of contexts c' such that $c \leq_C c'$), and, for $H_i = \langle c_1, c_2, \dots, c_h \rangle$, $\otimes(H_i)$ is shorthand for the expression $(\succ_{c_1} \otimes \succ_{c_2} \otimes \dots \otimes \succ_{c_h})$.

The result easily follows by unfolding Equation 1 and then applying the left-distributive property.

Example 7. For the context poset in Figure 1 consider the preferences in Table 1, which mimic those in Example 1.⁴ In context $c_1 = (\text{Italy})$, pasta is preferred to beef and beef to salad. On the other hand, in context $c_2 = (\text{Naples})$, pizza is preferred to both pasta and beef. It follows that in c_2 it is pasta \approx_{c_2} beef, thus such foods can be ordered using the preferences valid in c_1 , as shown

⁴ For simplicity in the table we just show the key of each tuple.

Table 1. Base and complete preferences under the \mathcal{CC} propagation for the context poset in Figure 1

c	\succ_c	$\succ_c^{+\mathcal{CC}}$
c_1	{pasta \succ beef, beef \succ salad, pasta \succ salad}	{pasta \succ beef, beef \succ salad, pasta \succ salad}
c_2	{pizza \succ pasta, pizza \succ beef}	{pizza \succ pasta, pizza \succ beef, pasta \succ beef}
c_3	{salad \succ pasta, salad \succ beef}	{salad \succ pasta, salad \succ beef, pasta \succ beef}
c_4	\emptyset	{pasta \succ beef}

in the right-most column of the table. Conversely, a salad is preferred to both pasta and beef in $c_3 = (\text{summer}; \text{Italy})$. In the most specific context $c_4 = (\text{summer}; \text{Naples})$, for which no base preferences are given, Pareto composition is applied to $\succ_{c_2}^{+\mathcal{CC}}$ and $\succ_{c_3}^{+\mathcal{CC}}$. These both agree on preferring pasta to beef, whereas other foods stay unordered due to the semantics of Pareto composition. Thus, the best alternatives in c_4 are pizza, pasta, and salad. This can be expressed by the PC-expression $\succ_{c_4} \otimes ((\succ_{c_2} \otimes \succ_{c_1}) \oplus (\succ_{c_3} \otimes \succ_{c_1}))$, which is the unfolding of Equation 1, or equivalently by the canonical PC-expression $(\succ_{c_4} \otimes \succ_{c_2} \otimes \succ_{c_1}) \oplus (\succ_{c_4} \otimes \succ_{c_3} \otimes \succ_{c_1})$.

In spite of the intuitive form of Eq. 1, we have the following negative result:

Theorem 1. *The \mathcal{CC} propagation is fair but not specific.*

Proof. To prove fairness, let c_1 and c_2 be two unordered contexts in $C[c]$, with $t_1 \succ_{c_1} t_2$ and $t_2 \succ_{c_2} t_1$, and let $t_1 \approx_{c_i} t_2 \forall c_i : (c \leq_C c_i <_C c_1) \vee (c \leq_C c_i <_C c_2)$. Let $c_{k,1}$ ($c_{k,2}$) be a context in $\text{COV}_C(c)$ such that $c <_C c_{k,1} \leq_C c_1$ ($c <_C c_{k,2} \leq_C c_2$, respectively). Due to the semantics of Pareto operator, either t_1 is still better than t_2 in the complete preferences in $c_{k,1}$, or the two tuples are incomparable in this context. Similar arguments hold for $c_{k,2}$, from which it is derived that t_1 and t_2 are incomparable in c . To see why \mathcal{CC} violates specificity, consider the poset in Figure 1 and any preference configuration BP such that: $t_1 \succ_{c_3} t_2$, $t_2 \succ_{c_1} t_1$, $t_1 \approx_{c_2} t_2$, and $t_1 \approx_{c_4} t_2$. To show that for no such BP it is $t_1 \succ_{c_4}^{+\mathcal{CC}} t_2$, thus violating specificity, it is enough to observe that $t_1 \succ_{c_3}^+ t_2$ and $t_2 \succ_{c_2}^+ t_1$, which, according to Pareto composition, leads to have t_1 and t_2 incomparable in c_4 .

Example 8. Let us slightly revise the base preferences in Table 1 assuming now that there are no base preferences in c_2 . By proceeding as in Example 7 it is still derived that $\succ_{c_4}^{+\mathcal{CC}} = \{\text{pasta} \succ \text{beef}\}$, since the complete preferences in c_2 coincide with those of c_1 while stay unchanged in c_3 . This contradicts the specificity principle, for which the complete preferences in c_4 should coincide with those in c_3 , thus making pizza and salad the only best alternatives.

4.2 The Active-Cover Propagation

Since Example 8 shows that contexts with no base preferences, such as c_2 , might invalidate the specificity property of the whole propagation process, it is interesting to study the effect of not considering at all such contexts. For this, given a preference configuration BP, let us say that a context c is *active* if $\text{BP}(c) \neq \emptyset$.

Table 2. Base and complete preferences under the \mathcal{AC} propagation for the context poset in Figure 1

c	\succ_c	$\succ_c^{+\mathcal{AC}}$
c_1	{pasta \succ beef, beef \succ salad, pasta \succ salad}	{pasta \succ beef, beef \succ salad, pasta \succ salad}
c_2	\emptyset	{pasta \succ beef, beef \succ salad, pasta \succ salad}
c_3	{salad \succ pasta, salad \succ beef}	{salad \succ pasta, salad \succ beef, pasta \succ beef}
c_4	\emptyset	{salad \succ pasta, salad \succ beef, pasta \succ beef}

The *Active Cover (AC) propagation* just considers the covering of c with respect to the poset $A \subseteq C$ of all active contexts, rather than C . More precisely, let $\text{COV}_A(c) = \{c_1, \dots, c_l\}$. Under the \mathcal{AC} propagation the complete preferences in context c , $\text{CP}_{\mathcal{AC}}(c) = \succ_c^{+\mathcal{AC}}$, are now computed as:

$$\succ_c^{+\mathcal{AC}} = \succ_c \otimes (\succ_{c_1}^{+\mathcal{AC}} \oplus \succ_{c_2}^{+\mathcal{AC}} \oplus \dots \oplus \succ_{c_l}^{+\mathcal{AC}}) \quad (3)$$

Note that if c itself is inactive, above Equation still correctly applies by considering the poset $A \cup \{c\}$.

Table 2 shows how Example 8 change under the \mathcal{AC} propagation, with the complete preferences in c_4 now consistent with what observed in the Example. However, in spite of being insensitive to the side-effects of contexts with no base preferences, even \mathcal{AC} cannot always guarantee specificity.

Theorem 2. *The \mathcal{AC} propagation is fair but not specific.*

Proof. Fairness follows from the same arguments used in the proof of Theorem 1. The same counterexample used in the proof of Theorem 1 applies here to show that \mathcal{AC} violates specificity, with the only additional hypothesis that $t_1 \approx_{c_2} t_2$, yet $\text{BP}(c_2) \neq \emptyset$ (i.e., c_2 is active).

4.3 The Tuple-Specific Cover Propagation

The rationale under the third propagation we introduce, called *Tuple-specific Cover (TC)*, is that the arguments used for discarding inactive contexts can be refined so as to drop, when comparing tuples t_1 and t_2 , also those contexts for which there is no specific preference relating t_1 and t_2 (thus, for each pair of tuples t_1 and t_2 , a specific subset of the active context poset A is considered).

Given a preference configuration BP, we say that a context c is (t_1, t_2) -active if $t_1 \not\approx_c t_2$. Let A_{t_1, t_2} be the poset of (t_1, t_2) -active contexts and $\text{COV}_{A_{t_1, t_2}}(c) = \{c_1, \dots, c_m\}$ be the covering of context c in the A_{t_1, t_2} poset. In the \mathcal{TC} propagation, tuples t_1 and t_2 are compared using the following equation, in which the observation that t_1 and t_2 are either ordered or incomparable in all contexts in $\text{COV}_{A_{t_1, t_2}}(c)$ is exploited to avoid recursion:

$$t_1 \succ_c^{+\mathcal{TC}} t_2 \iff t_1 [\succ_c \otimes (\succ_{c_1} \oplus \succ_{c_2} \oplus \dots \oplus \succ_{c_m})] t_2 \quad (4)$$

Example 9. Consider the poset in Figure 1 and the BP configuration:

$$\succ_{c_1} = \{t_2 \succ t_1, t_2 \succ t_3\} \quad \succ_{c_2} = \{t_1 \succ t_3, t_2 \succ t_3\} \quad \succ_{c_3} = \{t_1 \succ t_2, t_1 \succ t_3\} \quad \succ_{c_4} = \emptyset$$

It is $A_{t_1, t_2} = \{c_1, c_3\}$, $A_{t_2, t_3} = \{c_1, c_2\}$, and $A_{t_1, t_3} = \{c_2, c_3\}$. Thus, $\text{COV}_{A_{t_1, t_2}}(c_4) = \{c_3\}$, $\text{COV}_{A_{t_2, t_3}}(c_4) = \{c_2\}$, and $\text{COV}_{A_{t_1, t_3}}(c_4) = \{c_2, c_3\}$. According to both \mathcal{AC} and \mathcal{CC} propagation, tuples t_1 and t_2 are incomparable in context c_4 , since $\succ_{c_2}^{+\mathcal{CC}} = \succ_{c_2}^{+\mathcal{AC}}$ includes the preference $t_2 \succ t_1$, as inherited from c_1 , whereas $t_1 \succ t_2$ is an element of $\succ_{c_3}^{+\mathcal{CC}} = \succ_{c_3}^{+\mathcal{AC}}$. Instead, the \mathcal{TC} propagation *does not* consider context c_2 for ordering t_1 and t_2 , since c_2 is not (t_1, t_2) -active ($t_1 \approx_{c_2} t_2$), thus $t_1 \succ_{c_4}^{+\mathcal{TC}} t_2$. The complete preferences for all the three propagations in context c_4 are as follows (preferences coincides in the other contexts):

- $\succ_{c_4}^{+\mathcal{CC}} = \succ_{c_4}^{+\mathcal{AC}} = \{t_1 \succ t_3, t_2 \succ t_3\}$
- $\succ_{c_4}^{+\mathcal{TC}} = \{t_1 \succ t_2, t_2 \succ t_3, t_1 \succ t_3\}$

The following result shows that \mathcal{TC} can be indeed considered the “ultimate” semantics for preference propagation.

Theorem 3. *The \mathcal{TC} propagation is both fair and specific.*

Proof. Fairness stems directly from the definition of $\text{COV}_{A_{t_1, t_2}}(c)$. Specificity is also guaranteed, since when the three conditions of Definition 8 hold for any $c_1 <_C c_2$, $c_1, c_2 \in A[c]$, namely: $t_1 \succ_{c_1} t_2$, $t_2 \succ_{c_2} t_1$, and $t_1 \approx_{c_i} t_2 \forall c_i : c \leq_C c_i <_C c_1$, the definition of $\text{COV}_{A_{t_1, t_2}}(c)$ guarantees that $c_2 \notin \text{COV}_{A_{t_1, t_2}}(c)$. Thus, there exists a preference configuration BP for which it is $t_1 \succ_{c_j} t_2 \forall c_j \in \text{COV}_{A_{t_1, t_2}}(c)$, which implies $t_1 \succ_c^{+\mathcal{TC}} t_2$.

Apparently, the \mathcal{TC} propagation requires a distinct covering for each pair of tuples. However this is not true, since *there exists a PC-expression, the same for all pairs of tuples, that implements \mathcal{TC} propagation*. The intuition behind this result is that specificity needs to avoid that a preference $t_1 \succ t_2$, for which a conflicting preference exists in a more specific context, propagates along a chain that does not order t_1 and t_2 (which is the reason why both \mathcal{CC} and \mathcal{AC} violate specificity). Algebraically, this requires a PC-expression, which we denote $E_A^{\mathcal{TC}}(c)$, that is maximally “grouped on the right”, so that this pass-through phenomenon is inhibited. The following provides a formal definition of $E_A^{\mathcal{TC}}(c)$.

Definition 12 (PC-expression for \mathcal{TC} propagation). *Let c' be a context in $A[c]$ and let $\{c_1, \dots, c_k\}$ be the contexts in $A[c]$ that are covered by c' . The “right-grouped” expression $E_A^{\text{RG}}(c, c')$ is recursively defined as follows:*

$$\begin{cases} E_A^{\text{RG}}(c, c') = (E_A^{\text{RG}}(c, c_1) \oplus \dots \oplus E_A^{\text{RG}}(c, c_k)) \otimes \succ_{c'} \\ E_A^{\text{RG}}(c, c) = \succ_c \end{cases}$$

Let $\{\hat{c}_1, \dots, \hat{c}_n\}$ be the set of maximal elements in $A[c]$ (i.e., the contexts \hat{c}_i in $A[c]$ such that there is no context $c' \in A[c]$ for which $\hat{c}_i <_C c'$). Then:

$$E_A^{\mathcal{TC}}(c) = E_A^{\text{RG}}(c, \hat{c}_1) \oplus \dots \oplus E_A^{\text{RG}}(c, \hat{c}_n) \quad (5)$$

Example 10. Consider the poset in Figure 1, and assume that all contexts are active. The PC-expression $E_A^{\mathcal{TC}}(c_4)$ is $((\succ_{c_4} \otimes \succ_{c_2}) \oplus (\succ_{c_4} \otimes \succ_{c_3})) \otimes \succ_{c_1}$. For convenience, this can also be more compactly rewritten, by applying the left-distributive property of \otimes , as $\succ_{c_4} \otimes (\succ_{c_2} \oplus \succ_{c_3}) \otimes \succ_{c_1}$.

Intuitively, $E_A^{TC}(c)$ can be obtained from the canonical expression by first grouping chains on maximal elements and factoring them out, then recursively applying this process to the so-reduced chains until no more factors can be extracted.

Theorem 4. *The PC-expression $E_A^{TC}(c)$ correctly computes the TC propagation, i.e., $t_1 \succ_c^{+TC} t_2$ iff $t_1 E_A^{TC}(c) t_2$.*

Proof. If $t_1 \not\approx_c t_2$ the result is obvious, since in both cases only \succ_c is considered. Then, assume $t_1 \approx_c t_2$. We show that $E_A^{TC}(c)$ propagates downward to context c all and only the preferences concerning tuples t_1 and t_2 for those contexts $c_j \in \text{COV}_{A_{t_1, t_2}}(c)$, from which the result follows. If $c_j \in \text{COV}_{A_{t_1, t_2}}(c)$, it is $t_1 \not\approx_{c_j} t_2$ and $t_1 \approx_{c_i} t_2$ holds for any context c_i such that $c \leq_C c_i <_C c_j$. From the definition of $E_A^{RG}(c, c_j)$ it is immediate to derive that $E_A^{TC}(c)$ propagates the preference of c_j to c . By contradiction, assume now that $c_k \notin \text{COV}_{A_{t_1, t_2}}(c)$, $t_1 \not\approx_{c_k} t_2$, yet this preference is propagated by $E_A^{TC}(c)$ to context c . From the hypothesis, there exists a context $c_j \in \text{COV}_{A_{t_1, t_2}}(c)$ such that $c_j <_C c_k$. From the definitions of $E_A^{TC}(c)$ and $E_A^{RG}(c, c_k)$, it turns out that $E_A^{TC}(c)$ includes the sub-expression $(\dots \succ_{c_j} \dots) \otimes \succ_{c_k}$, and this is the case for every occurrence of \succ_{c_k} . Since the left operand includes \succ_{c_j} , for which t_1 and t_2 are not equivalent, the preference of c_k on these tuples does not propagate to c , proving the assert.

We conclude with a major result establishing a precise relationship among the three propagations we have analyzed so far.

Theorem 5. *Let c be a context in the context poset C . Then, the complete preferences in c under the three propagations, \mathcal{CC} , \mathcal{AC} and \mathcal{TC} , satisfy the following relationships: $\succ_c^{+CC} \subseteq \succ_c^{+AC} \subseteq \succ_c^{+TC}$ and $\approx_c^{CC} = \approx_c^{AC} = \approx_c^{TC}$.*

Proof (sketch). ($\approx_c^{CC} = \approx_c^{AC} = \approx_c^{TC}$): Due to the semantics of Pareto and Prioritized composition, two tuples t_1 and t_2 are equivalent according to \succ_c^+ iff this holds in all contexts whose base preferences appear in the PC-expression computing \succ_c^+ . These are clearly the same for \mathcal{AC} and \mathcal{TC} . Since the additional contexts used by \mathcal{CC} are all inactive, all tuples are equivalent in such contexts, from which the result follows.

($\succ_c^{+CC} \subseteq \succ_c^{+AC}$): Both propagations compute \succ_c^+ using a PC-expression that is equivalent to the canonical form, i.e., Pareto composition of all the “products” $\otimes(H_i)$, where H_i is a maximal chain (of the context poset C in the case of \mathcal{CC} , and of the active poset A in the case of \mathcal{AC}). If H_1 and H_2 are two chains, with $H_2 \subseteq H_1$, it is simple to see that $\otimes(H_1) \oplus \otimes(H_2) \subseteq \otimes(H_1)$. A repeated application of this yields the result, after observing that each $\otimes(H_i)$ in Equation 2 is equivalent to an expression $\otimes(H_i^-)$, $H_i^- \subseteq H_i$, obtained by discarding inactive contexts from H_i (since such contexts are uninfluential to the result of $\otimes(H_i)$). ($\succ_c^{+AC} \subseteq \succ_c^{+TC}$): The proof follows similar arguments.

Example 11. The preferences in Table 2 lead to have only one maximal chain in $A[c_4]$, i.e., $\langle c_3, c_1 \rangle$. Thus, $\succ_{c_4}^{+AC} = \succ_{c_3} \otimes \succ_{c_1}$. On the other hand, since the maximal chains in $C[c_4]$ are $\langle c_4, c_3, c_1 \rangle$ and $\langle c_4, c_2, c_1 \rangle$, yet c_2 and c_4 are inactive, we can equivalently consider the chains $\langle c_3, c_1 \rangle$ and $\langle c_1 \rangle$. It follows that $\succ_{c_4}^{+CC} = \succ_{c_1} \oplus (\succ_{c_3} \otimes \succ_{c_1}) \subseteq \succ_{c_3} \otimes \succ_{c_1} = \succ_{c_4}^{+AC}$.

5 Related Works

Preferences in databases have been extensively investigated in recent years following two main approaches: in the *quantitative* approach a numerical score is associated with tuples [2,10], while in the *qualitative* one a (strict) partial order relation is defined on tuples [6,8]. We have adopted the latter, which is more general from an order-theoretic point of view. Moreover, our approach is parametric with respect to the context model used since we only exploit the ability to relate different contexts according to a partial order between them, a feature common to the majority of the models proposed [4].

Recently, a number of papers have focused on the management of contextual preferences [1,14,15,16,18]. The main difference with the present paper is that all of them follow a pragmatic approach based on specific heuristics and then focus on implementation issues. More in detail, the works by van Bunnigen et al. [18] and Agrawal et al. [1] do not address the issue of combining preferences defined in different contexts. Agrawal et al. [1] introduce a technique for the management of contextual preferences, based on the qualitative approach, but they do not consider any generalization relationship between contexts. Stefanidis and Pitoura [16] consider quantitative preferences in a hierarchical context model. Preferences in a context c are computed from preferences defined in context c' that generalizes c and is at “minimal distance” from c in the hierarchy. With respect to the propagation properties introduced in Section 3, this approach is specific, but it is not fair and does not respect the propagation scope. Miele et al. [14] also consider numerical preferences and distances between contexts, but preferences defined on contexts at a distance from c that is not minimal are also considered, provided they are not *overwritten* by some other preference. Since a smaller context distance does not imply $c_1 \leq_C c_2$, this approach respects the propagation scope, it is specific, but it is not fair.

Mindolin and Chomicki [13] have recently introduced *p-skylines*, a particular case of PC-expressions in which each preference relation (i) is used only once, and (ii) is a total order over an attribute of interest. Taken together, these two restrictions simplify the problem of determining equivalence and containment of expressions, but this comes at the price of a reduced expressive power. In particular, p-skyline expressions (i) cannot be used for arbitrary context posets, and (ii) limits the kind of preferences we can define.

Contextual preferences could be considered as a particular case of *conditional preference networks* (CP-nets), a tool largely investigated in the AI field [3] and used for database querying [7,12]. Behind the surface, there are important differences between our work and that on CP-nets. With CP-nets one defines, for each attribute of interest, a set of total orders that are conditionally dependent on some other attribute(s). The resulting preferences are then defined as the transitive closure of the union of such orders [12] (this might not be an order since cycles might arise). Conversely, we start with a set of arbitrary strict partial orders and study how to compose them in a context poset, ensuring that the result is always a strict partial order.

6 Conclusions

In this paper we have considered the problem of how database preferences propagate when they depend on the context of the user. Unlike previous approaches, which are based on heuristic arguments, we have tackled the problem in a principled way and have proposed an algebraic model for expressing preference propagation, based on the well-known Pareto and Prioritized composition rules. We have then analyzed with this model three alternative propagation methods and shown that one of them satisfies all the desirable propagation properties.

As future work we plan to investigate specialized optimization techniques for contextual preference queries on large databases and to study how our results apply to numerical preference models.

References

1. Agrawal, R., Rantzaou, R., Terzi, E.: Context-Sensitive Ranking. In: SIGMOD, pp. 383–394 (2006)
2. Agrawal, R., Wimmers, E.L.: A Framework for Expressing and Combining Preferences. In: SIGMOD, pp. 297–306 (2000)
3. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Art. Intell. Research* 21, 135–191 (2004)
4. Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F.A., Tanca, L.: A Data-Oriented Survey of Context Models. *SIGMOD Record* 36(4), 19–26 (2007)
5. Bolchini, C., Curino, C., Orsi, G., Quintarelli, E., Rossato, R., Schreiber, F.A., Tanca, L.: And what Can Context Do for Data? *Com. of ACM* 52(11), 136–140 (2009)
6. Chomicki, J.: Preference Formulas in Relational Queries. *ACM Trans. on Database Systems* 28(4), 427–466 (2003)
7. Ciaccia, P.: Querying Databases with Incomplete CP-nets. In: M-PREF (2007)
8. Kießling, W.: Foundations of Preference in Database Systems. In: VLDB, pp. 311–322 (2005)
9. Kießling, W.: Preference Queries with SV-Semantics. In: COMAD, pp. 15–26 (2005)
10. Koutrika, G., Ioannidis, Y.E.: Personalization of Queries in Database Systems. In: ICDE, pp. 597–608 (2004)
11. Li, X., Feng, L., Zhou, L.: Contextual Ranking of Database Querying Results: A Statistical Approach. In: Roggen, D., Lombriser, C., Tröster, G., Kortuem, G., Havinga, P. (eds.) EuroSSC 2008. LNCS, vol. 5279, pp. 126–139. Springer, Heidelberg (2008)
12. Mindolin, D., Chomicki, J.: Hierarchical CP-nets. In: M-PREF (2007)
13. Mindolin, D., Chomicki, J.: Discovering Relative Importance of Skyline Attributes. In: VLDB (2009)
14. Miele, A., Quintarelli, E., Tanca, L.: A Methodology for Preference-Based Personalization of Contextual Data. In: EDBT, pp. 287–298 (2009)
15. Stefanidis, K., Pitoura, E., Vassiliadis, P.: Adding Context to Preferences. In: ICDE, pp. 846–855 (2007)
16. Stefanidis, K., Pitoura, E.: Fast contextual preference scoring of database tuples. In: EDBT, pp. 344–355 (2008)
17. Torlone, R., Ciaccia, P.: Which Are My Preferred Items? In: RPEC, pp. 217–225 (2002)
18. van Bunnigen, A.H., Fokkinga, M.M., Apers, P.M.G., Feng, L.: Ranking Query Results Using Context-Aware Preferences. In: DBRank, pp. 269–276 (2007)