Bipolar Queries and Relative Object Qualification in Scope of User-Assisted Database Querying

Mateusz Dziedzic*†, Guy De Tré*, Janusz Kacprzyk† and Sławomir Zadrożny†
*Dept. of Telecommunication and Information Processing
Ghent University, Ghent, Belgium
Email: {Mateusz.Dziedzic, Guy.Detre}@ugent.be
†Systems Research Institute
Polish Academy of Sciences, Warsaw, Poland
Email: {Mateusz.Dziedzic, Janusz.Kacprzyk, Slawomir.Zadrozny}@ibspan.waw.pl

***Abstract*—Two similar approaches to the modeling of *bipolar user preferences*, namely *bipolar queries* (and their extension to *contextual bipolar queries*) and *queries with relative object qualification*, are presented from the point of view of a *user-assisted* database querying. Their close relation is discussed, similarities and differences highlighted and possible disadvantages for the user studied. Then a direct, practical comparison of both approaches supported by computational examples on a simplified, yet realistic data set is presented and discussed and possible implementation and usage recommendations are made.**

***Index Terms*—Bipolar queries, Bipolar operators, Contextual bipolar queries, Flexible queries, Fuzzy logic, Linguistic queries, Relative object qualification**

I. Introduction

A conventional database query (composed with e.g. an SQL query language) against a conventional, relational database specifies precise (crisp) values, intervals or inequalities for properties of items sought, e.g. "find all *'semi-detached'* houses *'located in Greater Glasgow'* with *'2 to 4 bedrooms'* and *'price below £200k'* which can be expressed as:

```
SELECT * FROM houses
WHERE type = 'semi-detached'
AND location = 'Greater Glasgow'
AND (bedrooms BETWEEN 2 AND 4)
AND price <= 200000.
```

However, many times our intentions are not that well specified and a possibility to use an imprecise, more human-centric approach expressed in natural language would be welcomed. As an example, above SQL query could be restated as:

> "Find all *'semi-detached'* houses located *'close to Glasgow'* with *'moderate number of bedrooms'* and *'affordable price'*". (1)

Such queries can be formulated using fuzzy sets theory leading to introduction of *fuzzy linguistic queries* (dating back to late 1970s, see e.g. [1] for references).

The next step in our journey to the human-centric database querying, *bipolar queries*, originates in a seminal work of Lacroix and Lavency [2] and was formalized and further extended, among others, by Yager [3], [4], Bordogna and Pasi [5], Chomicki [6] (who proposed to name them as *queries with preferences*) or Zadrożny and Kacprzyk [7]–[11]. The main goal of such queries is to express the complex user preferences modeled separately as *positive* and *negative* conditions (or combinations of such conditions), with the complement of the latter to be mandatory fulfilled and the former being of a secondary importance that can be treated as user's preference (cf. another approach to bipolarity adopted by Bosc, Pivert et al. in e.g. [12], [13], where a lexicographic ordering is used). For example, following our housing query (1), houses sought might have to be located within the specified area due to e.g. work commute, where *'not within Greater Glasgow'* or *'far from Glasgow'* becomes the mandatory *negative* condition, and only then the preferred price is taken into consideration forming the *positive* condition:

> "Find all houses located *'close to Glasgow'* and possibly *'inexpensive'*". (2)

The characteristic feature of our approach (see e.g. [7]–[11]) to *bipolar queries* is the reference to the possibility or impossibility (the *"and possibly"* operator above) of satisfying the *positive* condition determined over the whole data under consideration. For example, to evaluate the satisfaction degrees of Equation 2 one has to determine first whether there are any tuples in the whole data which satisfy both conditions simultaneously. Thus, for every tuple the possibility (impossibility) condition gives the same intermediary result. However, it is not hard to imagine situations in which one would like to assess the possibility (impossibility) within some context of the tuple, e.g., with respect to other tuples sharing some common attribute with the one under consideration. Continuing our housing example, such *contextual bipolar query* (as referred to in e.g. [14]–[16]) can be formulated as follows:

> "Find all houses that are *not too small* and possibly, within the context of their *'location'*, *'affordable'*", (3)

where the *affordability* of each property is evaluated within its own neighborhood, which could possibly lead to finding some local market gems.

Another similar approach to database querying with roots in the necessity of evaluating unequally important conditions was introduced by Tudorie et al. [17], [18] as (queries with) *relative*

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

object qualification, referred here as *ROQ* for brevity, and later expanded to (queries with) *relative object qualification to group on another attribute* (*grouped ROQ*) in [19], [20]. The main difference between these two and the *(contextual) bipolar queries* lies within the way how the two conditions are combined: Tudorie et al. proposed to use the results of the *primary* condition to formulate a linguistic (sub)domain for the *secondary* one. Translating (2) and (3) to this approach one would obtain the following queries:

$$\text{"Find all } \textit{'inexpensive'} \text{ houses among ones located } \textit{'close to Glasgow'}\text{"; and} \tag{4}$$

$$\text{"Find all } \textit{'not too small'} \text{ houses that are } \textit{'affordable'} \text{ among ones within } \textit{'similar location'}\text{"}, \tag{5}$$

discussed and computationally evaluated later in this paper. Please note the *primary* and *secondary* conditions correspond to the *negative* and *positive* ones of aforementioned *bipolar queries* with the *among* operator providing a different formalization of the bipolar aggregation, hence following our understanding of bipolarity.

To conclude this section we would like to advocate a *user-assisted*, or *interactive*, approach to both database querying and summarization, on which our research is based (see e.g. [21], [22] for general introduction by Kacprzyk and Zadrożny in the scope of data summarization, to which presented here *bipolar* approach can also be applied [16], [23], or [24]–[26] for some details of their *FQUERY for Access* package). Let us bring back our natural language based housing query (1). The fuzzy predicates used in it, i.e. *'close to Glasgow'*, *'moderate number of bedrooms'* and *'affordable price'*, can have very different meanings for each individual user and assuming their form without "consultation" with them feels impractical.

The rest of this paper is structured as follows: in section II we introduce, briefly discuss and provide examples of *bipolar queries*, including their *contextual* extension; in section III we do the same for Tudorie et al's *queries with relative object qualification*; in section IV we discuss the relation between these two approaches to *bipolarity* providing examples of analogous queries; and in section V we discuss selected characteristics and computational evaluation of three sample groups of queries to support our observations. We conclude the paper with a short general discussion and summary.

II. Bipolar Queries

As explained in section I, *bipolar queries* form a natural extension of *fuzzy linguistic queries* and their role is to express the complex user preferences formulated as *positive* and *negative* conditions. The crucial component of such queries is the *bipolar operator* connecting both sets of conditions. In our previous works (see e.g. [7]–[11], [14], [15]) two such operators were introduced: the *"and possibly"* and *"or if impossible"* operators, along with an extension of the approach itself, the *contextual bipolar queries*, which we will discuss further in this section.

A. Bipolar "And Possibly" Operator

Following aforementioned works of Yager [3], [4], Bordogna and Pasi [5] or Zadrożny and Kacprzyk [7]–[11], we model the required and desired conditions using the *"and possibly"* operator, with the bipolar query taking the form of:

$$C \text{ and possibly } P, \tag{6}$$

which may be exemplified, referring to our housing example, by:

$$\text{"}\textit{located close to Glasgow}\text{ and possibly }\textit{affordable}\text{"}. \tag{7}$$

Following [2] we adopt a crisp interpretation of 6 as the following logical formula:

$$C \text{ and possibly } P \equiv C \wedge (\exists s(C_s \wedge P_s) \Rightarrow P) \tag{8}$$

where, for brevity, C and C_s denotes $C(t)$ and $C(s)$, respectively (the same for P). Formula (8) can then be directly fuzzified (cf. Dubois and Prade [27] and Zadrożny and Kacprzyk [7], [8] for studies on other concepts of fuzzification of bipolar queries) as:

$$\mathcal{T}(C \text{ and possibly } P) = \min(C, \max(1 - \max_{s \in R} \min(C_s, P_s), P)) \tag{9}$$

where, for brevity, R denotes the whole data (relation) and C and C_s denotes $\mu_C(t)$ and $\mu_C(s)$, respectively (the same for P).

An interpretation of the proposition (6) given by (8) makes it true (to a high degree) for a tuple t only if either of the two conditions holds:

1) It satisfies (to a high degree) both conditions C and P; or
2) It satisfies C and there is no other tuple in the *whole database* which satisfies both conditions simultaneously. (10)

The value of:

$$\max_{s \in R} \min(C_s, P_s) \tag{11}$$

from (9), expressing the fulfillment degree of $\exists s(C_s \wedge P_s)$ (also denoted as $\exists CP$), may be interpreted as the measure of interference between C and P; the lower its value the harder it is to simultaneously satisfy both conditions.

The general fuzzification approach adopted in (9) left open the choice of representation of logical connectives and quantifiers, which were later studied by Zadrożny and Kacprzyk in [9], [11] (see also [28] for a brief overview in the scope of bipolar linguistic summaries). In the above, the standard minimum and maximum operators were employed along with the related S-implication due to their distributivity and idempotency properties.

B. Bipolar "Or if Impossible" Operator

In our subsequent works ([15], [16]) another *bipolar* operator, *"or if impossible"* was introduced. Although our further discussion in this paper will not be based on it, let us introduce it here for completeness. The *"or if impossible"* operator closely follows the approach of the *"and possibly"* operator with its satisfaction degree being dependent on the whole data under consideration, but this time it favors the *positive* condition P:

$$P \text{ or if impossible } C, \tag{12}$$

which may be exemplified by:

"*'affordable'* or if impossible *'located close to Glasgow'*". (13)

and formalized in the crisp form following discussion from (8) with:

$$P \text{ or if impossible } C \equiv P \vee (\neg \exists s P_s \wedge C) \tag{14}$$

where P, C and P_s denotes $P(t)$, $C(t)$ and $P(s)$, respectively, and further fuzzified (see (9) for reference) as:

$$\mathcal{T}(P \text{ or if impossible } C) = \max(P, \min(1 - \max_{s \in R} P_s, C)). \tag{15}$$

Proposition (12) interpreted as (15) is true (to a high degree) for each tuple t if either of the following two conditions holds:

1) It satisfies (to a high degree) condition P; or
2) It satisfies C and there is no other tuple in the *whole database* which satisfies condition P.

(16)

All other remarks from subsection II-A, including the interpretation of the formula $\exists s P_s$ as a measure of the possibility of satisfaction of the *preference* condition P, holds.

C. Contextual Bipolar Queries

As described in the previous two subsections, the possibility and impossibility of meeting each respective conditions in *"and possibly"* and *"or if impossible"* operators are evaluated over the whole data under consideration, which is limiting in many situations and one might want to assess the possibility (or impossibility) of satisfying relevant parts of the operators within some specifies context of the tuple, thus allowing for even more flexible querying. This idea was explored and developed by Zadrożny et al. in [14]–[16] where *contextual bipolar queries*, an extension of the bipolar queries described in the previous section, originated.

The novel part of such queries is the *context* $W(t, s)$, defined as a relation between the tuple t under consideration and every other tuple in the data, which allows for a flexible definition of the subset of tuples that are relevant or similar to the tuple under consideration, e.g. (using our housing examples) *'located close to the same city'*, *'similarly appraised'* or even *'located close to each other'*, which can be used to simultaneously query "local gems" of undervalued properties using their own neighborhoods as references.

Following the structure of previous two subsections we will first re-introduce an example from section I and its interpretation followed by crisp and fuzzified formulas. Let us imagine a potential user with an unlimited budget (for the simplicity of this example, but one can think of e.g. a business reseller looking for attractive "investment opportunities") who is looking for *'not too small'*, but *'affordable' 'within their neighborhoods'*, houses while hunting for some great deals without really worrying where exactly they will be located:

"Find all houses that are *'not too small'* and possibly, within the context of their *'location'*, *'affordable'*". (17)

Such query can be modeled as a proposition:

$$C \text{ and possibly } P \text{ with regard to } W, \tag{18}$$

and interpreted in the crisp case as:

$$C \text{ and possibly } P \text{ with regard to } W \equiv C \wedge (\exists s(W_{t,s} \wedge C_s \wedge P_s) \Rightarrow P), \tag{19}$$

where C, P, $W_{t,s}$, C_s and P_s denote, respectively, $C(t)$, $P(t)$, $W(t, s)$, $C(s)$ and $P(s)$; and after fuzzification as:

$$\mathcal{T}(C \text{ and possibly } P \text{ with regard to } W) = \min(C, \max(1 - \max_{s \in R} \min(W_{t,s}, C_s, P_s), P)) \tag{20}$$

It is worth highlighting that *contextual bipolar queries* follow all observations made in subsection II-A and subsection II-B.

III. Queries with Relative Object Qualification

Another possible approach to *bipolarity* in database querying, although not referred to as such directly by the authors, was introduced by Tudorie et al. [17], [18] as (queries with) *relative object qualification* (*ROQ*), and later expanded to (queries with) *relative object qualification to group on another attribute* (*grouped ROQ*) in [19], [20].

A. Relative Object Qualification

Tudorie et al. introduced an aggregation operator *"among"* with the query formulated as:

$$P \text{ among } C, \tag{21}$$

which might be exemplified by the following "housing" query:

"*affordable* among ones *located close to Glasgow*". (22)

which evaluation boils down to limiting the data under consideration to tuples "pre-selected" by the *primary* condition (i.e. where $\mu_{primary}(t) > 0$) to formulate a linguistic (sub)domain for the *secondary* one.

Discussed behavior of the *"among"* operator let us directly compare it to our *"if possible"* operator, where similar approach is assumed for (the negation of) the *negative* condition - the *positive* condition is only taken into account when the *negative* one is fulfilled (to a high degree). This parallel let us

Algorithm 1.

1) Evaluate *primary* query conditions and select only tuples with $\mu_{primary}(t) > 0$;
2) Modify subdomains of *secondary* conditions based on subsample of tuples from step 1;
3) Transform predicates from original domains to subdomains from step 2 (see also discussion in subsection III-A);
4) Evaluate *secondary* query conditions;
5) Calculate the final query satisfaction degree (e.g. using a conjunction of intermediary satisfaction degrees from steps 1 and 4).

consider the *"among"* operator as a *bipolar* one, albeit not in the most straightforward understanding of it.

Algorithm 1 presents a simplified query evaluation procedure (please refer to any of [17]–[20] for a more comprehensive discussion), but for clarity we will briefly explain the predicate transformation (Step 3) here as being crucial to the evaluation of discussed operator.

Let us use a *Price* attribute defined over $[a, b]$ by n linguistic values (e.g. *low*, *medium* and *high*) spanning across its whole domain as an example here. After Step 2 of Algorithm 1 a subdomain $[a', b']$ of *Price* is obtained where $a <= a'$ and $b' <= b$. Then, a scaling and shifting transformation $f : [a, b] \to [a', b']$ is applied to the membership functions of aforementioned linguistic values such that $\mu'_P(x) = f \circ \mu_P(x)$ and consequently:

$$\mu'_P(x) = \mu_P(a + \frac{b-a}{b'-a'}(x - a')),$$

resulting in a uniform coverage of the new subdomain $[a', b']$ of *Price* with analogical n linguistic values (e.g. *low'*, *medium'* and *high'*). Again, please refer to any of [17]–[20] for more details.

B. ROQ to Group on Another Attribute

The second type of queries introduced by Tudorie et al. corresponds roughly with our idea of *context*, but in the original implementation (see [19], [20]) is limited to an unary relation $G(t)$[1], which limitations and possible extension will be briefly explored in later sections.

Although Tudorie et al. do not introduce additional conditions, which would directly correspond to our *context*, taking our *bipolar contextual query* example (17) as a departure point, one could think of the following query:

> "Find all houses that are *'not too small'* and *'affordable'* among ones *'located in each of the unique areas given in the data*[2]*'*". (23)

to which examples and formulas from [19], [20] could easily be extended, and formulate a query proposition:

$$C \text{ and } P \text{ among } G. \tag{24}$$

[1]Some notation details follow our previous works and are not reflected in works of Tudorie et al.

The main drawback of the *queries with grouped ROQ* seems to be the ("hidden" from the user) mapping of domains and predicates (see steps 2 and 3 of the Algorithm 1), which we will illustrate and discuss in the following sections.

IV. Relation between Bipolar Queries and Relative Object Qualification

Although Tudorie et al. do not directly refer to *queries with ROQ* as *bipolar*, the specific approach to user preferences (discussed in section III), where some of them are of a clearly higher importance than others, and the latter are evaluated relative to them, suggests common roots of the interpretation of bipolarity for both *bipolar queries* and *queries with ROQ*, which we will explore here followed by some computational evaluation later on.

Examples and discussion from section II and section III let us formulate the following two pairs of closely related queries:

1) *Bipolar queries* and *queries with ROQ*; and
2) *Contextual bipolar queries* and *queries with grouped ROQ*,

discussed further in subsection IV-A and subsection IV-B, respectively.

A. Bipolar Queries and Relative Object Qualification

Let us consider queries consisting of the following fuzzy predicates:

- Negation of the negative condition C (i.e. a mandatory one): *'located close to Glasgow'*; and
- Positive condition P (a preference): *'inexpensive'*. (25)

Using the above mentioned predicates one can build the following two queries:

1) *Bipolar query* (C and possibly P) in the form of:

   > "Find all houses *'located close to Glasgow'* and possibly *'inexpensive'*", (26)

   which then could be "translated" to:
2) *Query with ROQ* (P among C):

   > "Find all *'inexpensive'* houses among ones *'located close to Glasgow'*"; (27)

roughly maintaining the desired search characteristics of looking for houses *'located close to Glasgow'*, but also *'inexpensive'*, if it is possible at all.

The main difference between these two queries comes from the way the two conditions are combined:

1) In *bipolar queries* the secondary conditions are taken into account only when it is actually possible to satisfy them and the mandatory ones simultaneously (see (8) and (9) and the rest of section II); while

[2]Mentioned areas are each considered individually here, de facto forming a set of separate ROQ queries.

2) In *queries with ROQ* they are evaluated only for the subset of records with predicates modified according to Algorithm 1 (section III), which guarantees their high satisfaction degrees for at least some of the tuples, see examples 2 and 3 from section V (please note this does not ensure high satisfaction degrees for the whole query due to conjunction operators).

The approach employed in the latter has an interesting property which might be considered both positive and negative depending on user intentions and context. Namely, it disregards satisfaction degrees of the *positive* condition for the unmodified predicates — this introduces, possibly hidden from the user, a side effect of the query where predicates defined by them (following the *user-assisted* approach, as advocated by us at the end of section I and forming the central stipulation of this paper) are modified. On one hand, this "scaling" of satisfaction degrees might form a good indicator that "nothing better" can be found in the data, and, on the other, it artificially overstates them, where low satisfaction degrees could give the user additional information on how difficult are both conditions to simultaneously fulfill.

Examples of the above observation are presented on Figure 2 and Figure 3 of section V where predicate:

$$inexp(t) : Price \rightarrow [0,1] = \text{trap}(0, 0, \mathbf{100}, \mathbf{200}) \tag{28}$$

is "silently" modified to:

$$inexp(t) : Price \rightarrow [0,1] = \text{trap}(0, 0, \mathbf{89}, \mathbf{146}); \tag{29}$$

and predicate:

$$exp(t) : Price \rightarrow [0,1] = \text{trap}(\mathbf{200}, \mathbf{300}, 400, 400) \tag{30}$$

to:

$$exp(t) : Price \rightarrow [0,1] = \text{trap}(\mathbf{146}, \mathbf{202}, 400, 400), \tag{31}$$

both of which quite drastically change satisfaction degrees of houses priced in the range of £150k–200k (technically, the whole domain of *Price* is reduced, but this does not impact this observation).

B. Contextual Bipolar Queries and Relative Object Qualification to Group on Another Attribute

To reformulate queries (26) and (27) into examples of *contextual* and *grouped ROQ* we need to introduce a third set of predicates:

- Context W of the *contextual bipolar query*: *'located close to the same city'*; and
- Grouping condition G of the *grouped ROQ*: *'located close to each of the unique cities given in the data'* (see footnote in (23)). (32)

Using the preference condition P from (25) (mandatory condition C is omitted for simplicity, which does not impact reported results) and the context and grouping conditions introduced above, the two sample queries can be formulated as follows:

1) *Contextual bipolar query* (possibly P wrt W):
"Find all *'inexpensive'* houses wrt ones *'located close to the same city'*"; and (33)
2) *Grouped ROQ* query (P among G):
"Find all *'inexpensive'* houses among ones *'located close to each of the cities'*". (34)

Besides observations from subsection IV-A, the two queries above differ in how the *context* and *grouping conditions* are defined. As a brief reminder, the *context* $W(t, s)$ is defined as a relation between the tuple t under consideration and every other tuple in the data, which allows for a flexible definition of the subset of tuples that are relevant or similar to the tuple under consideration, while the *grouping condition* $G(t)$, as defined in [19], [20], acts more like a consecutive "treading" through some tuple properties, see sample queries above with *context* $W(t, s)$ defined as *'located close* ***to the same city****'* and *grouping condition* $G(t)$ as *'located close* ***to each of the cities****'*. See subsection II-C and subsection III-B for more details.

It is worth highlighting that, to the best of our knowledge, nothing, besides maybe a potential computational complexity, stands against redefining the *grouping condition* as a relation $G(t, s)$ following our definition of the *context* — with such amendment the two types of queries discussed in this section would be even more closely related.

V. Computational Evaluation and Discussion

As a reminder, to illustrate the similarities and differences discussed in the previous two sections we will use the following three situations where the user searches for:

1) *'Inexpensive'* houses *'located close to Glasgow'*, where the former predicate forms the *positive* condition (a *preference*) and the latter the *negative* (*mandatory*) one — to introduce a basic *bipolar* approach to querying, refer to (26) and (27) for actual queries used and subsection V-A for discussion;
2) *'Expensive'* ones, also *'located close to Glasgow'* — a simple modification of the ones above to better show how Algorithm 1 impacts the *optional* predicate's definition, discussed in subsection V-B; and
3) All *'inexpensive'* houses, irrespective to their location, but taking it into account while evaluating the property's "inexpensiveness" — as an example of *contextual*, or *grouped*, approach, as exemplified by queries (33) and (34) and discussed in subsection V-C.

In our numerical examples the following predicates will be used (see Figure 1 for visual representation):

$$\begin{aligned} close(t) : Dist &\rightarrow [0,1] = \text{trap}(0, 0, 5, 10) \\ inexp(t) : Price &\rightarrow [0,1] = \text{trap}(0, 0, 100, 200) \\ exp(t) : Price &\rightarrow [0,1] = \text{trap}(200, 300, 400, 400) \end{aligned} \tag{35}$$

and the Table I presents the sample data with three attributes, *Area*, *Distance* from the "city center" (as explained in the table

Table I
HOUSE PRICES AROUND EDINBURGH AND GLASGOW.

#	Area	Distance[a]	Price [£k]
1	Edinburgh	13.9	142
2	Edinburgh	11.2	160
3	Edinburgh	10.6	168
4	Edinburgh	10.6	200
5	Edinburgh	7.3	210
6	Edinburgh	5.2	245
7	Edinburgh	4.1	255
8	Edinburgh	10.5	265
9	Edinburgh	4.6	315
10	Edinburgh	4.8	330
11	Glasgow	5.8	75
12	Glasgow	9.7	100
13	Glasgow	14.2	120
14	Glasgow	29.7	135
15	Glasgow	4.0	140
16	Glasgow	6.3	176
17	Glasgow	7.6	210
18	Glasgow	8.8	219
19	Glasgow	27.5	260
20	Glasgow	10.3	275

[a] Shortest driving distance in miles to the Edinburg and Glasgow City Chambers as found on www.google.pl/maps.

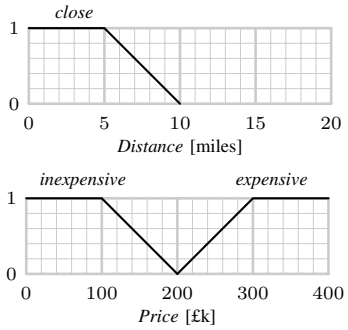

Figure 1. Predicates of *Distance* and *Price* with their membership functions (defined with (35)).

note) in miles and *Price* in £k, found on a real estate website. The following three sections will discuss the results obtained.

A. Computational Example 1

This example presents a basic *bipolar* approach to querying and illustrates the impact of Algorithm 1 on returned satisfaction degrees of queries (26) and (27). As can be observed, "preselection" of tuples by *ROQ* results in a reduced domain of *Price* and "left-sqeezed" *inexpensive* predicate (see Figure 2 and details in (28) and (29)), which lead to (as presented in Table II) generally much lower values of $\mu_{inexpensive}$ and, finally, returning less tuples (only 3/10 as compared to 6/10) with lower satisfaction degrees for some of the returned ones.

To the contrary, because the value of $\exists CP$ expressed as $(\max_{s\in R} \min(C_s, P_s))$ equals 0.84 (i.e. it is not fully possible to satisfy both conditions simultaneously, so the ones not satisfying the *preference* one receive a partial "relief" from

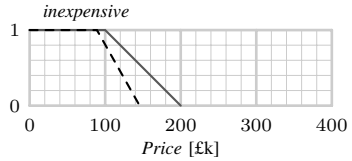

Figure 2. Predicate *inexpensive* in its original form (gray solid line) and modified by Step 3 from Tudorie's Algorithm 1 (dashed).

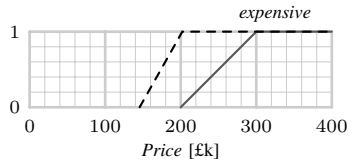

Figure 3. Predicate *expensive* in its original form (gray solid line) and modified by Step 3 from Tudorie's Algorithm 1 (dashed).

that), if only tuples satisfy the *mandatory* condition, their overall query satisfaction degree cannot be lower than $1-0.84$ (property $\#12$ is pretty far from Glasgow's center, hence the lower score).

B. Computational Example 2

In this analogical example we are still using the same queries as above, i.e. (26) and (27), just with an inverse of the positive condition: the user is looking for an *expensive* house within the Glasgow area. Here we observe even larger modification of the domain of *Price* and (this time "right-sqeezed") *expensive* predicate (for details of this modification see Figure 3 and (30) and (31)).

It is worth highlighting that because most houses in our data are not *expensive* in the *bipolar query* example most of the tuple "goodness" comes from the *mandatory* condition with the satisfaction degree just slightly reduced by the $\exists CP$ expression (see Table III); and for *queries with ROQ* some tuples evaluate to $\mu_{inexpensive}(t) = 1$ as designed. The decision whether this is a desired "side-effect" will vary by use case, although in this example values of $\mathcal{D}(t)$ seem to be more appropriate.

C. Computational Example 3

In this last (even more simplified) example of *contextual bipolar queries* and *queries with grouped ROQ* (see queries (33) and (34)) we wanted to focus on two main aspects of them:

- Both approaches let us evaluate a single query with *context* or *grouping conditions* for multiple "groups" at once (although, technically, some steps are performed either tuple by tuple or over their pre-selected grouping); and
- By removing the *mandatory* condition, we could even better illustrate the effect Algorithm 1 has on the data (sub)domains and predicates of the query.

Obtained results are given in Table IV and details of domain and predicate modification are omitted, but their effects are clearly visible in the data — please note the "stretching"

Table II
RECORDS FROM TABLE I WITH SATISFACTION DEGREES FOR QUERIES (26) AND (27) DISCUSSED IN SUBSECTION V-A.

#	Area	Dist.	Price	μ_{close}[a]	$\mu_{inexpensive}$[a]	$\mathcal{D}$[b]	$\mathcal{T}$[b]	Δ[c]
11	Glasgow	5.8	75	0.84	1.00 (1.00)	0.84	0.84	0.00
15	Glasgow	4.0	140	1.00	0.60 (0.10)	0.60	0.10	0.50
16	Glasgow	6.3	176	0.74	0.24 (0.00)	0.24	0.00	(—)
17	Glasgow	7.6	210	0.48	0.00 (0.00)	0.16	0.00	(—)
18	Glasgow	8.8	219	0.24	0.00 (0.00)	0.16	0.00	(—)
12	Glasgow	9.7	100	0.06	1.00 (0.81)	0.06	0.06	0.00

[a] μ_{close} is the same for both queries, hence it is reported only once, while $\mu_{inexpensive}$ forming the *secondary preference* condition of the query, is evaluated differently for each; *ROQ* evaluation is reported within the brackets.
[b] $\mathcal{D}$ and $\mathcal{T}$ denote, respectively, the *bipolar* and *query with ROQ* satisfaction degrees.
[c] (—) denote tuples returned by only one of the queries.

Table III
RECORDS FROM TABLE I WITH SATISFACTION DEGREES FOR MODIFIED QUERIES (26) AND (27) DISCUSSED IN SUBSECTION V-B.

#	Area	Dist.	Price	μ_{close}[a]	$\mu_{expensive}$[a]	$\mathcal{D}$[a]	$\mathcal{T}$[a]	Δ[a]
11	Glasgow	5.8	75	0.84	0.00 (0.00)	0.81	0.00	(—)
15	Glasgow	4.0	140	1.00	0.00 (0.00)	0.81	0.00	(—)
16	Glasgow	6.3	176	0.74	0.00 (0.54)	0.74	0.54	0.20
17	Glasgow	7.6	210	0.48	0.10 (1.00)	0.48	0.48	0.00
18	Glasgow	8.8	219	0.24	0.19 (1.00)	0.24	0.24	0.00
12	Glasgow	9.7	100	0.06	0.00 (0.00)	0.06	0.00	(—)

[a] See notes in Table II.

effect on $\mu_{inexpensive}(t)$ where for both groups (Edinburgh and Glasgow) the new membership degrees are covering the whole $[0, 1]$ interval. Again, the decision whether this effect is desirable will vary for every use-case.

In addition, the effect of $\exists CP$ expression when the simultaneous satisfaction of both *negative* and *positive* conditions is not possible is again clearly visible in the Edinburgh part of the table where most tuples were evaluated to $1 - 0.58 = 0.42$ as 0.58 was deemed as the highest satisfaction degree for both conditions simultaneously.

VI. CONCLUSIONS

In this work we first introduced two possible approaches to a very appealing subject of modeling of *bipolar user preferences* in database querying: *bipolar queries* and *queries with relative object qualification*.

Then, we discussed relation between them highlighting some similarities and differences in intentions and interpretation in the scope of *user-assisted* database querying, where Tudorie et al's approach revealed one major, possibly negative characteristic, namely it "silently" modifies (sub)domains of data and query predicates of the *positive/secondary/preference* conditions.

The main contribution of this research is the direct comparison of both approaches supported by some computational examples which will be extended towards more thorough theoretical examination and will also cover *linguistic data summarization* as a natural extension in our planned future research.

Table IV
RECORDS FROM TABLE I WITH SATISFACTION DEGREES FOR QUERIES (33) AND (34) DISCUSSED IN SUBSECTION V-C.

#	Area	Dist.	Price	$\mu_{inexpensive}$[a]	$\mathcal{D}$[a]	$\mathcal{T}$[a]	Δ[a]
1	Edinburgh	13.9	142	0.58 (1.00)	0.58	1.00	-0.42
2	Edinburgh	11.2	160	0.40 (1.00)	0.42	1.00	-0.58
3	Edinburgh	10.6	168	0.32 (0.90)	0.42	0.90	-0.45
4	Edinburgh	10.6	200	0.00 (0.46)	0.42	0.46	-0.04
5	Edinburgh	7.3	210	0.00 (0.33)	0.42	0.33	0.09
6	Edinburgh	5.2	245	0.00 (0.00)	0.42	0.00	(—)
7	Edinburgh	4.1	255	0.00 (0.00)	0.42	0.00	(—)
8	Edinburgh	10.5	265	0.00 (0.00)	0.42	0.00	(—)
9	Edinburgh	4.6	315	0.00 (0.00)	0.42	0.00	(—)
10	Edinburgh	4.8	330	0.00 (0.00)	0.42	0.00	(—)
11	Glasgow	5.8	75	1.00 (1.00)	1.00	1.00	0.00
12	Glasgow	9.7	100	1.00 (0.93)	1.00	0.93	0.07
13	Glasgow	14.2	120	0.80 (0.68)	0.80	0.68	0.12
14	Glasgow	29.7	135	0.65 (0.49)	0.65	0.49	0.16
15	Glasgow	4.0	140	0.60 (0.42)	0.60	0.42	0.18
16	Glasgow	6.3	176	0.24 (0.00)	0.24	0.00	(—)

[a] See notes in Table II.

REFERENCES

[1] J. Kacprzyk and A. Ziółkowski, "Database queries with fuzzy linguistic quantifiers," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 3, pp. 474–479, 1986.
[2] M. Lacroix and P. Lavency, "Preferences: Putting more knowledge into queries," in *Proceedings of the 13 International Conference on Very Large Databases*, Brighton, UK, 1987, pp. 217–225.
[3] R. R. Yager, "Higher structures in multi-criteria decision making," *International Journal of Man-Machine Studies*, vol. 36, pp. 553–570, 1992.
[4] ——, "Fuzzy logic in the formulation of decision functions from linguistic specifications," *Kybernetes*, vol. 25, no. 4, pp. 119–130, 1996.
[5] G. Bordogna and G. Pasi, "Linguistic aggregation operators of selection criteria in fuzzy information retrieval," *International Journal of Intelligent Systems*, vol. 10, no. 2, pp. 233–248, 1995.
[6] J. Chomicki, "Querying with intrinsic preferences," *Lecture Notes in Computer Science*, vol. 2287, pp. 34–51, 2002.
[7] S. Zadrożny, "Bipolar queries revisited," in *Modeling Decisions for Artificial Intelligence. MDAI 2005*, ser. Lecture Notes in Computer Science, V. Torra, Y. Narukawa, and S. Miyamoto, Eds. Berlin, Heidelberg: Springer, 2005, vol. 3558, pp. 387–398.
[8] S. Zadrożny and J. Kacprzyk, "Bipolar queries and queries with preferences (invited paper)," in *17th International Workshop on Database and Expert Systems Applications (DEXA '06)*, Kraków, Poland, 2006, pp. 415–419.
[9] ——, "Bipolar queries using various interpretations of logical connectives," in *Foundations of Fuzzy Logic and Soft Computing. IFSA 2007*, ser. Lecture Notes in Computer Science, P. Melin, O. Castillo, L. T. Aguilar, J. Kacprzyk, and W. Pedrycz, Eds. Berlin, Heidelberg: Springer, 2005, vol. 4529, pp. 181–190.
[10] ——, "Bipolar queries: An approach and its various interpretations," in *Proc. of the Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference (IFSA/EUSFLAT 2009)*, Lisbon, Portugal, 2009, p. 1288–1293.
[11] ——, "Bipolar queries: An aggregation operator focused perspective," *Fuzzy Sets and Systems*, vol. 196, pp. 69–81, 2012.
[12] P. Bosc, O. Pivert, A. Mokhtari, and L. Liétard, "Extending relational algebra to handle bipolarity," in *Proceedings of the 2010 ACM Symposium on Applied Computing*. New York, NY, USA: Association for Computing Machinery, 2010, p. 1718–1722.
[13] P. Bosc and O. Pivert, "On a fuzzy bipolar relational algebra," *Inf. Sci.*, vol. 219, p. 1–16, 2013.
[14] S. Zadrożny, J. Kacprzyk, M. Dziedzic, and G. de Tré, "Contextual bipolar queries," in *Advance Trends in Soft Computing*, ser. Studies in Fuzziness and Soft Computing, M. Jamshidi, V. Kreinovich, and

J. Kacprzyk, Eds. Springer International Publishing, 2014, vol. 312, pp. 421–428.

[15] S. Zadrożny, J. Kacprzyk, and M. Dziedzic, "Contextual bipolar queries: "or if impossible" operator case," in *2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15)*, ser. Advances in Intelligent Systems Research, J. M. Alonso, H. Bustince, and M. Reformat, Eds., vol. 89, Gijón, Spain, 2015, pp. 1266–1273.

[16] ——, "On a new type of contextual queries and linguistic summaries of a bipolar type," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2015, pp. 1–8.

[17] C. Tudorie, S. Bumbaru, and C. Segal, "New kind of preference in database fuzzy querying," in *Proc. of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, IPMU 2006*, Paris, France, 2006, p. 1389–1395.

[18] C. Tudorie, S. Bumbaru, and L. Dumitriu, "Relative qualification in database flexible queries," in *2006 3rd International IEEE Conference Intelligent Systems, IS 2006*, London, UK, 2006, pp. 83–88.

[19] C. Tudorie, "Qualifying objects in classical relational database querying," in *Handbook of Research on Fuzzy Information Processing in Databases*, J. Galindo, Ed. London, UK: IGI Global, 2008, pp. 218–245.

[20] C. Tudorie and D. Ştefănescu, "Special cases of relative object qualification: Using the among operator," in *Intelligent Systems and Technologies*, ser. Studies in Computational Intelligence, J. W. Horia-Nicolai Teodorescu and L. C. Jain, Eds. Berlin, Heidelberg: Springer-Verlag, 2009, vol. 217, pp. 181–191.

[21] J. Kacprzyk and S. Zadrożny, "Data mining via linguistic summaries of data: an interactive approach," in *Methodologies for the Conception, Design and Application of Soft Computing. Proceedings of IIZUKA'98*, T. Yamakawa and G. Matsumoto, Eds., Iizuka, Japan, 1998, pp. 668–671.

[22] ——, "Data mining via linguistic summaries of databases: an interactive approach," in *A New Paradigm of Knowledge Engineering by Soft Computing*, L. Ding, Ed. Singapore: World Scientific, 2001, pp. 325–345.

[23] M. Dziedzic, J. Kacprzyk, S. Zadrożny, and G. de Tré, "Quantified quality criteria of contextual bipolar linguistic summaries," in *Challenging Problems and Solutions in Intelligent Systems. Studies in Computational Intelligence*, ser. Studies in Computational Intelligence, G. de Trė, P. Grzegorzewski, J. Kacprzyk, J. W. Owsiński, W. Penczek, and S. Zadrożny, Eds. Springer, Cham, 2016, vol. 634, pp. 421–428.

[24] J. Kacprzyk and S. Zadrożny, "FQUERY for Access: fuzzy querying for a windows-based DBMS," in *Fuzziness in Database Management Systems*, P. Bosc and J. Kacprzyk, Eds. Heidelberg: Physica-Verlag, 1995, pp. 415–433.

[25] ——, "The paradigm of computing with words in intelligent database querying," in *Computing with Words in Information/Intelligent Systems. Part 1. Foundations. Part 2. Applications*, L. A. Zadeh and J. Kacprzyk, Eds. Heidelberg and New York: Springer-Verlag, 1999, pp. 382–398.

[26] ——, "Computing with words in intelligent database querying: standalone and internet-based applications," *Information Sciences*, vol. 134, no. 1-4, pp. 71–109, 2001.

[27] D. Dubois and H. Prade, "Bipolarity in flexible querying," in *FQAS 2002*, ser. LNAI, T. Andreasen, A. Motro, H. Christiansen, and H. L. Larsen, Eds. Berlin, Heidelberg: Springer-Verlag, 2002, vol. 2522, pp. 174–182.

[28] M. Dziedzic, J. Kacprzyk, and S. Zadrożny, "Bipolar linguistic summaries: A novel fuzzy querying driven approach," in *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*. Edmonton, Canada: IEEE, 2013, pp. 1279–1284.