Similarity Group-by Operators for Multi-dimensional Relational Data (Extended Abstract)

Mingjie Tang[1] Ruby Y. Tahboub[1] Walid G. Aref[1] Mikhail J. Atallah[1] Qutaibah M. Malluhi[2]
Mourad Ouzzani[3] Yasin N. Silva[4]
[1]Purdue University [2]Qatar University [3]Qatar Computing Research Institute [4]Arizona State University
{tang49,rtahboub,aref,mja}@cs.purdue.edu, qmalluhi@qu.edu.qa, mouzzani@qf.org.qa, ysilva@asu.edu

***Abstract*—The SQL group-by operator plays an important role in summarizing and aggregating large datasets in a data analytics stack. The Similarity SQL-based Group-By operator (SGB, for short) extends the semantics of the standard SQL Group-by by grouping data with similar but not necessarily equal values. While existing similarity-based grouping operators efficiently realize these approximate semantics, they primarily focus on one-dimensional attributes and treat multi-dimensional attributes independently. However, correlated attributes, such as in spatial data, are processed independently, and hence, groups in the multi-dimensional space are not detected properly. To address this problem, we introduce two new SGB operators for multi-dimensional data. The first operator is the clique (or distance-to-all) SGB, where all the tuples in a group are within some distance from each other. The second operator is the distance-to-any SGB, where a tuple belongs to a group if the tuple is within some distance from any other tuple in the group. Since a tuple may satisfy the membership criterion of multiple groups, we introduce three different semantics to deal with such a case: (i) eliminate the tuple, (ii) put the tuple in any one group, and (iii) create a new group for this tuple. We implement and test the new SGB operators and their algorithms inside PostgreSQL. The overhead introduced by these operators proves to be minimal and the execution times are comparable to those of the standard Group-by. The experimental study, based on TPC-H and a social check-in data, demonstrates that the proposed algorithms can achieve up to three orders of magnitude enhancement in performance over baseline methods developed to solve the same problem.**

I. Introduction

We introduce new similarity-based group-by operators that group multi-dimensional data using various metric distance functions. More specifically, we propose two SGB operators, namely SGB-All and SGB-Any, for grouping multi-dimensional data. SGB-All forms groups such that a tuple or a data item, say o, belongs to a group, say g, if o is at a distance within a user-defined threshold from all other data items in g. In other words, each group in SGB-All forms a clique of nearby data items in the multi-dimensional space. For example, all the two-dimensional points (a-e) in Figure 1a are within distance 3 from each other and hence form a clique. They are all reported as members of one group as they are all part of the output of SGB-All. In contrast, SGB-Any forms groups such that a tuple or a data item, say o, belongs to a group, say g, if o is within a user-defined threshold from at least one other data item in g. For example, all the two dimensional points in Figure 1b form one group. Point a is within Distance 3 from Point c that in turn is within Distance 3 from Points b, d, and f. Furthermore, Point e is within Distance 3 from Point d, and so on. Therefore, Points a-h of Figure 1b are reported as members of one group as part of the output.

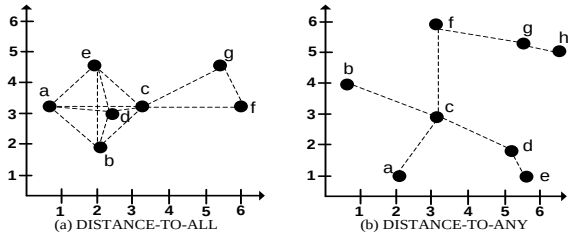


Fig. 1: The Semantics of Similarity predicates $\epsilon = 3$.

A. Similarity Group-By ALL (SGB-All)

Given a set of tuples whose grouping attributes form a set, say P, of two-dimensional points, where $P = \{p_1, ..., p_n\}$, the SGB-All operator $\check{\mathcal{G}}_{all}$ forms a set, say G_m, of groups of points from P such that $\forall g \in G_m$, the similarity predicate $\xi_{\delta,\epsilon}$ is TRUE for all pairs of points $\langle p_i, p_j \rangle \in g$, and g is maximal, i.e, there is no group g' such that $g \subseteq g'$. More formally,

$$\check{\mathcal{G}}_{all} = \{g \mid \forall p_i, p_j \in g,\ \xi_{\delta,\epsilon}(p_i, p_j) \wedge g\ is\ maximal\}$$

Figure 1 gives an example of two groups (a-e) and (c,f,g), where all pairs of elements within each group are within a distance $\epsilon \leq 3$. The proposed SQL syntax for the SGB-All operator is as follows:

```
SELECT column, aggregate-func(column)
FROM table-name
WHERE condition
GROUP BY column DISTANCE-TO-ALL [L2 | LINF]
WITHIN ϵ
ON-OVERLAP [JOIN-ANY | ELIMINATE |
FORM-NEW-GROUP]
```

SGB-All uses the following clauses to realize similarity-based grouping:

DISTANCE-TO-ALL: specifies the distance function to be applied by the similarity predicate when deciding the membership of points within a group.

- L2: L_2 (Euclidean distance).
- LINF: L_∞ (Maximum infinity distance)

ON-OVERLAP: is an arbitration clause to decide on a course of action when a data point is within Distance ϵ from more than one group. When a point, say p_i, matches the

membership criterion for more than one group, say $g_1 \cdots g_w$, one of the three following actions are taken:

- JOIN-ANY: the data point p_i is randomly inserted into any one group out of $g_1 \cdots g_w$.
- ELIMINATE: discard the data point p_i, i.e., all data points in the overlapping set.
- FORM-NEW-GROUP: insert p_i into a separate group, i.e., form new groups out of the points in $Oset$.

B. Similarity Group-By Any (SGB-Any)

Given a set of tuples whose grouping attributes from a set, say P, of two dimensinal points, where $P = \{p_1, ..., p_n\}$, the SGB-Any operator $\check{\mathcal{G}}_{any}$ clusters points in P into a set of groups, say G_m, such that, for each group $g \in G_m$, the points in g are all connected by edges to form a graph, where an edge connects two points, say p_i and p_j, in the graph if they are within Distance ϵ from each other, i.e,. $\xi_{\delta,\epsilon}(p_i, p_j)$. More formally,

$$\check{\mathcal{G}}_{any} = \{g \mid \forall p_i, p_j \in g, (\xi_{\delta,\epsilon}(p_i, p_j) \vee (\exists\, p_{k1}, ..., p_{kn},\ \xi_{\delta\epsilon}(p_i, p_{k1}) \wedge ... \wedge \xi_{\delta\epsilon}(p_{kn}, p_j))) \wedge g\ is\ maximal\}$$

The notion of distance-to-any between elements within a group is illustrated in Figure 1b, where $\epsilon = 3$. All of the points (a-h) form one group. The corresponding SQL syntax of the SGB-Any operator is as follows:

```
SELECT column, aggregate-func(column)
FROM table-name
WHERE condition
GROUP BY column DISTANCE-TO-ANY [L2 | LINF]
WITHIN ε
```

SGB-Any uses the DISTANCE-TO-ANY predicate that applies the metric space function while evaluating the distance between adjacent points. When using the semantics for SGB-Any, the case for points overlapping multiple groups does not arise. The reason is that when an input point overlaps multiple groups, the groups merge to form one large group.

II. Performance Evaluation and Discussion

We realize the proposed SGB operators inside PostgreSQL, the implementation can be accessed in [1]. The experiments are based on the TPC-H benchmark[1], and two real-world social check-in datasets, namely Brightkite[2] and Gowalla[3]. The experiments evaluate the effect of similarity threshold ϵ and data size on the SGB-All variants and SGB-ANY [2].

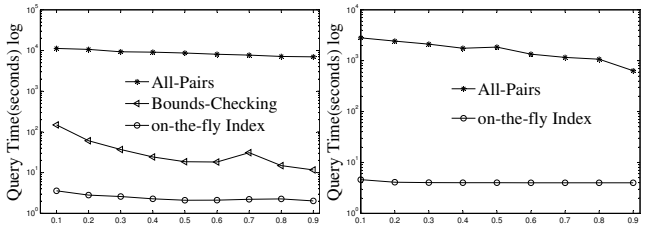
In Figure 2, we observe that the runtime of SGB-All JOIN-ANY decreases as the value of ϵ approaches 0.9. The experiment illustrates that the runtime for *All-Pairs* SGB-Any decreases as the value of ϵ increases. Furthermore, the runtime of the *on-the-fly Index* method [2] slightly changes. As a result, the speedup between the *All Pairs* and the *on-the-fly Index* methods slightly decreases.

In Figure 3, we observe that, as the data size increases, the runtime of the *All-Pairs* method increases quadratically, while the runtime of the *on-the-fly Index* method has a linear speedup. Moreover, the speedup results in the figure demonstrate that the *on-the-fly Index* method is approximately three orders of magnitude faster than *All-Pairs SGB-Any* as the data size increases.

[1]http://www.tpc.org/tpch/

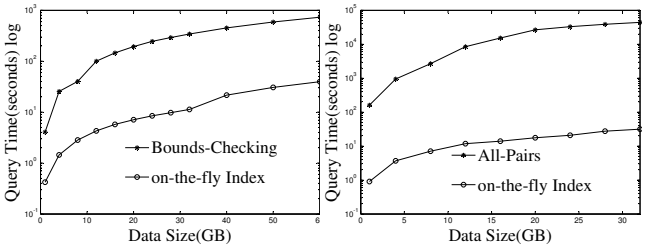
[2]https://snap.stanford.edu/data/loc-brightkite.html

[3]https://snap.stanford.edu/data/loc-gowalla.html



(a) SGB-All:JOIN-ANY (b) SGB-ANY

Fig. 2: The effect of similarity threshold ϵ on the SGB-All JOIN-ANY and SGB-ANY



(a) SGB-All:JOIN-ANY (b) SGB-ANY

Fig. 3: The effect of increasing data size on the SGB-All JOIN-ANY and SGB-ANY

III. Conclusion and Acknowledgment

In this paper, we address the problem of similarity-based grouping over multi-dimensional data. We define new similarity grouping operators with a variety of practical and useful semantics to handle overlap. The performance of SGB-All performs up to three orders of magnitude better than the naive *All-Pairs* grouping method. Moreover, the performance of the optimized SGB-Any performs more than three orders of magnitude better than the naive approach. Finally, the performance of the proposed SGB operators is comparable to that of standard relational Group-by.

This work was supported by an NPRP grant 4-1534-1-247 from the Qatar National Research Fund and by the National Science Foundation Grants IIS 0916614, IIS 1117766, and IIS 0964639.

References

[1] https://github.com/merlintang/sgb

[2] Tang, M., Tahboub, R., Aref, W., Atallah, M., Malluhi, Q., Ouzzani, M., Silva, Y. Similarity Group-by Operators for Multi-dimensional Relational Data. TKDE, vol. no. 99,.2015