Similarity Group-by Operators for Multi-dimensional Relational Data

Mingjie Tang1, Ruby Y. Tahboub1, Walid G. Aref1, Mikhail J. Atallah1, Qutaibah M. Malluhi2, Mourad Ouzzani3, Yasin Silva4

1Purdue University, 2Qatar University, 3Qatar Computing Research Institute, 4Arizona State University

Motivation
- Similarity search is everywhere, so is searching for database elements that are similar or close to a given query element.
- There is a need to group n-dimensional data tuples together that have similar (≈) values.
- We need to extend the SQL Group-By operator to support similarity-based grouping.

Semantics of Similarity Group-By (SGB)
- Given 2D data tuples T, and distance parameter ε, return groups of tuples from T that satisfy the predefined distance predicates: Distance-to-All (SGB-All), Distance-to-Any (SGB-Any)
- Distance-to-All: All the tuples in a group are within certain distance threshold ε from each other
- Distance-to-Any: A tuple belongs to a group if the tuple is within distance ε from any other tuple in the group
- ON-OVERLAP: To decide on a course of action when a point p is within Distance ε from more than one group.
- Possible actions: ON-OVERLAP JOIN-ANY: Data point p is inserted into any one of the overlapping groups.
- ON-OVERLAP JOIN-MULTI: Data point p overlaps more than one group.
- ON-Overlap FORM-NEW-GROUP: Insert p into a separate new group that contain all the overlapping points.

Example Queries
- Table Mobile Devices: (MDID, Latitude, Longitude) maintains the geographic locations of mobile devices

Application of SGB-Any:
- Identify groups of connected mobile devices using signal range as a similarity grouping threshold

SELECT ST_Polygon (Device lat, Device long) FROM MobileDevices
GROUP BY Device lat, Device long
DISTANCE-TO-LY L2 WITHIN SignalRange
ON-OVERLAP JOIN-ANY

Application of SGB-All-Form-New-Groups:
- Identify gateway devices (member of multiple groups)
- A gateway device acts as an entrance from one group to other

SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-ID, Device-long
DISTANCE-TO-AU L2 WITHIN SignalRange
ON-OVERLAP FORM-NEW-GROUP

Application of SGB-All-Join-Any:
- Identify devices that can communicate with each other directly based on their own signal strength

SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-id, Device-long
DISTANCE-TO-AU L2 WITHIN SignalRange
ON-OVERLAP JOIN-ANY

Application of SGB-All-Eliminate:
- Identify devices that cannot serve as a gateway, and devices from different group that cannot communicate without a gateway

SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-ID, Device-long
DISTANCE-TO-AU L2 WITHIN SignalRange
ON-OVERLAP ELIMINATE

Query Optimization
- SGB-All: Bounding-rectangle, convex hull, spatial index, disk-based hash tables
- SGB-Any: Spatial index, union-find, disk-based hash tables

Implementation
- Parser
- Rewriter
- Planner
- Executor

- Developed inside PostgreSQL 8.2
  - > 8k lines of code
  - Uses an in-memory R-tree index inside query executor
  - Memory protection
  - Transaction consistency
  - Fault recovery

Experiments
- Tested using TPC-H and social network check-in dataset (Gowalla, Brightkite)
- SGB operators implemented inside PostgreSQL 8.2.4
- Code is available at https://github.com/merlintang/sgb
- Tested query performance against straightforward realization of SGB, various other cluster algorithms, and standard Group-by of PostgreSQL

Related work
- Data cluster algorithms
  - Developed on top of the DBMS
  - Takes the DBMS as a black box
  - Suffers from the extraneous I/O due to impedance mismatch with data in the DB
- Similarity query processing algorithms
  - Well studied, but no previous work on multi-dimensional similarity-group-by

ACKNOWLEDGEMENTS
This work is supported by ONR Grant No. N00014-1-247 and National Science Foundation under Grant III-1117766.

Motivation
- Similarity search is everywhere, so is searching for database elements that are similar or close to a given query element.
- There is a need to group n-dimensional data tuples together that have similar (≈) values.
- We need to extend the SQL Group-By operator to support similarity-based grouping.

Semantics of Similarity Group-By (SGB)
- Given 2D data tuples T, and distance parameter ε, return groups of tuples from T that satisfy the predefined distance predicates: Distance-to-All (SGB-All), Distance-to-Any (SGB-Any)
- Distance-to-All: All the tuples in a group are within certain distance threshold ε from each other
- Distance-to-Any: A tuple belongs to a group if the tuple is within distance ε from any other tuple in the group
- ON-OVERLAP: To decide on a course of action when a point p is within Distance ε from more than one group.
- Possible actions: ON-OVERLAP JOIN-ANY: Data point p is inserted into any one of the overlapping groups.
- ON-OVERLAP JOIN-MULTI: Data point p overlaps more than one group.
- ON-Overlap FORM-NEW-GROUP: Insert p into a separate new group that contain all the overlapping points.

Example Queries
- Table Mobile Devices: (MDID, Latitude, Longitude) maintains the geographic locations of mobile devices

Application of SGB-Any:
- Identify groups of connected mobile devices using signal range as a similarity grouping threshold

SELECT ST_Polygon (Device lat, Device long) FROM MobileDevices
GROUP BY Device lat, Device long
DISTANCE-TO-LY L2 WITHIN SignalRange
ON-OVERLAP JOIN-ANY

Application of SGB-All-Form-New-Groups:
- Identify gateway devices (member of multiple groups)
- A gateway device acts as an entrance from one group to other

SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-ID, Device-long
DISTANCE-TO-AU L2 WITHIN SignalRange
ON-OVERLAP FORM-NEW-GROUP

Application of SGB-All-Join-Any:
- Identify devices that can communicate with each other directly based on their own signal strength

SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-id, Device-long
DISTANCE-TO-AU L2 WITHIN SignalRange
ON-OVERLAP JOIN-ANY

Application of SGB-All-Eliminate:
- Identify devices that cannot serve as a gateway, and devices from different group that cannot communicate without a gateway

SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-ID, Device-long
DISTANCE-TO-AU L2 WITHIN SignalRange
ON-OVERLAP ELIMINATE

Query Optimization
- SGB-All: Bounding-rectangle, convex hull, spatial index, disk-based hash tables
- SGB-Any: Spatial index, union-find, disk-based hash tables

Implementation
- Parser
- Rewriter
- Planner
- Executor

- Developed inside PostgreSQL 8.2
  - > 8k lines of code
  - Uses an in-memory R-tree index inside query executor
  - Memory protection
  - Transaction consistency
  - Fault recovery

Experiments
- Tested using TPC-H and social network check-in dataset (Gowalla, Brightkite)
- SGB operators implemented inside PostgreSQL 8.2.4
- Code is available at https://github.com/merlintang/sgb
- Tested query performance against straightforward realization of SGB, various other cluster algorithms, and standard Group-by of PostgreSQL

Related work
- Data cluster algorithms
  - Developed on top of the DBMS
  - Takes the DBMS as a black box
  - Suffers from the extraneous I/O due to impedance mismatch with data in the DB
- Similarity query processing algorithms
  - Well studied, but no previous work on multi-dimensional similarity-group-by

ACKNOWLEDGEMENTS
This work is supported by ONR Grant No. N00014-1-247 and National Science Foundation under Grant III-1117766.