Krajča P., Vychodil V.: Query Optimization Strategies in Similarity-Based Databases

Department of Computer Science, Palacky University, Olomouc (17. listopadu 12, CZ–77146 Olomouc, Czech Republic)

Introduction

Our Goal

Create a database system

:: based on the generalization of the Codd's relational model

:: corresponding to the theoretical model,

:: allowing similarity based queries (find a car which costs about $10,000),

:: allowing for implicit query optimizations,

:: supporting fast and efficient query processing.

Data Model: Basics

:: complete residuated lattice $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \to, 0, 1\rangle$, serves as a **scale of truth degrees**

:: nonempty **set** Y **of attributes** – names of columns

:: finite subset $R \subseteq Y$ is called a **relation scheme** (heading of the table)

:: each attribute $y \in Y$ has its **domain** D_y (set of attribute's values)

:: **cartesian product** of domains D_y ($y \in R$), denoted by $\prod_{y\in R} D_y$, is a set of all maps $r\colon R \to \bigcup_{y\in R} D_y$ such that $r(y) \in D_y$ for all $y \in R$

:: each $r \in \bigcup_{y\in R} D_y$ shall be called a **tuple** on R over domains D_y ($y \in R$)

:: **ranked data table** on R (shortly, an RDT) over domains D_y is any map

$$\mathcal{D}\colon \prod_{y\in R} D_y \to L$$

such that there are at most finitely many tuples r such that $\mathcal{D}(r) > 0$.

:: the degree $\mathcal{D}(r)$ assigned to tuple r by $\mathcal{D}$ shall be called a **rank** of tuple r in $\mathcal{D}$

:: RDTs are counterparts to the ordinary data tables in the original Codd's model

:: RDTs represent stored data and results of similarity-based queries where tuples are allowed to match conditions to degrees

Similarity Based Restriction

:: having a scale of truth degrees, L each domain D_y can be equipped with a map $\approx_y\colon D_y \times D_y \to L$, called a **similarity**, satisfying conditions of reflexivity and symmetry:

(i) $\approx_y(u, u) = 1$ for all $u \in D_y$;

(ii) $\approx_y(u, v) = \approx_y(v, u)$ for all $u, v \in D_y$

:: $u \approx_y v$ is interpreted as a degree to which $u \in D_y$ is similar to $v \in D_y$

Restriction

:: for an RDT $\mathcal{D}$ on R, attribute $y \in R$, and $d \in D_y$ similarity based restriction $\sigma_{y\approx d}(\mathcal{D})$ is defined by

$$(\sigma_{y\approx d}(\mathcal{D}))(r) = \mathcal{D}(r) \otimes (r(y) \approx_y d)$$

:: if $\mathcal{D}$ is result of query Q, the rank given by restriction is a degree to which "r matches Q and its y-values is similar to d.

:: $\mathcal{D}(r)$ and $(r(y) \approx_y d)$ can be seen as two sub-queries which are aggrgated by $\otimes$

:: one may use Fagin's algorithm to retrieve k tuples with highest ranks

:: Fagin's algorithm requires

- random access
- sorted access (tuples are listed in the descending order w.r.t. their ranks)

Example

	name	*price*	*type*	*year*
1.00	Ford Focus	9811.0	Hatchback	2011
1.00	Honda Accord	10600.0	Wagon	2010
1.00	Ford Fiesta	11560.0	Wagon	2011
1.00	Hyundai i30	11699.0	Hatchback	2010
1.00	BMW X5	12500.0	SUV	2004

Table 1. Ranked Data Table $cars$

	name	*price*	*type*	*year*
0.94	Ford Fiesta	11560.0	Wagon	2011
0.80	Hyundai i30	11699.0	Hatchback	2010
0.10	Honda Accord	10600.0	Wagon	2010

Table 2. Restriction $\sigma_{price\approx_{price}11500}(cars)$

Domains of Ordinal Data

Definition 1. Let $\lhd$ be a strict total order on D. Similarity $\approx\colon D \times D \to L$ is called monotone with respect to $\lhd$ if, for all elements $d_1, d_2, d_3 \in D$ such that $d_1 \lhd d_2 \lhd d_3$ we have $d_1 \approx d_3 \leq d_1 \approx d_2 \wedge d_2 \approx d_3$.

Remark 1. For subsets of real numbers, one can define monotone similarity w.r.t. genuine strict ordering of reals by computing absolute difference between values, and map the difference to L using antitone scaling function

Proposition 1. Let $f\colon \mathbb{R} \to \mathbb{R}$ be injective and monotone (or antitone) and let $s\colon \mathbb{R} \to L$ be antitone. Then, $\approx$ defined by

$$d_1 \approx d_2 = s\big(|f(d_1) - f(d_2)|\big) \tag{1}$$

is monotone w.r.t. $<$.

Listing Tuples

:: assuming an index (B-tree) over attributes $y_1, \ldots, y_n$,

:: domain values $d_1 \in D_{y_1}, \ldots, d_n \in D_{y_n}$,

:: monotone similarity operator $\approx_{y_n}$,

:: one can efficiently retrieve tuples with the highest ranks for queries:

$$(y_1 = d_1) \otimes \cdots \otimes (y_{n-1} = d_{n-1}) \otimes (y_n \approx_{y_n} d_n). \tag{2}$$

Sketch of the Algorithm

1. two cursors – one moving forward, one moving backward
2. at the beginning cursors are set to a record (or to its closest neighbor) which satisfies:

$$y_1 = d_1,\ \ldots,\ y_{n-1} = d_{n-1},\ \text{and}\ y_n = d_n$$

3. backward moving cursor moves one step backward
4. condition (2) is evaluated for tuples pointed by both cursors
5. tuple with the highest degree is returned and the given cursor moves to the next record

:: algorithm also applicable for $<$, $\leq$, $\geq$, $between$, or $(y \approx_y d) \vee (y \leq_y d)$

Domains of Nominal Data

:: we can represent similarities of nominal values as an RDT

:: for a finite domain D_y we can consider an RDT $\mathcal{D}_y$ on relation scheme $\{y, y'\}$ over domains $D_y = D_{y'}$ such that

$$\mathcal{D}_y = r(y) \approx_y r(y') \tag{3}$$

:: i.e., rank represents similarity of two values

Listing Tuples

:: for an RDT $\mathcal{D}_y$ representing similarities of values from D_y and RDT $\mathcal{D}$ with attribute y and value $d \in D_y$

:: similarity based restriction may be reduced to a natural join and projection as follows:

$$\sigma_{y\approx d}(\mathcal{D}) = \mathcal{D} \bowtie \pi_{\{y\}}(\sigma_{y'\approx d}(\mathcal{D}_y)) \tag{4}$$

Projection

:: for an RDT $\mathcal{D}$ on T is the projection $\pi_R(\mathcal{D})$ of $\mathcal{D}$ onto $R \subseteq T$ defined as follows:

$$(\pi_R(\mathcal{D}))(r) = \bigvee\big\{\mathcal{D}(rs) \mid s \in \textstyle\prod_{y\in T\setminus R} D_y\big\} \tag{5}$$

for each tuple $r \in \prod_{y\in R} D_y$.

Natural Join

:: $\mathcal{D}_1 \bowtie \mathcal{D}_2$ is a relation on $R \cup S \cup T$, consisting of concatenation of all joinable tuples rs and st from $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, such that

$$(\mathcal{D}_1 \bowtie \mathcal{D}_2)(rst) = \mathcal{D}_1(rs) \otimes \mathcal{D}_2(st), \tag{6}$$

Natural Join: Sketch of the Algorithm

Sorted access phase
Retrieve tuples from $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, in the descending order according to their ranks. Denote the sets of retrieved tuples by R_1 and R_2, respectively. Continue with enlarging R_1 and R_2 until

$$J = \{\langle rs, st\rangle \mid rs \in R_1 \text{ and } st \in R_2\} \tag{7}$$

has at least k elements.

Random access phase
For each $rs \in R_1$ retrieve all values $\mathcal{D}_2(st) > 0$ and store rst;
For each $st \in R_2$ retrieve all values $\mathcal{D}_1(rs) > 0$ and store rst.

Computation phase
Take the set S of all tuples rst stored in the previous step, sort the set according to $\mathcal{D}_1(rs) \otimes \mathcal{D}_2(st)$ in the descending order and output first k records.

International Center for Information and Uncertainty
SSIU 2014: International Spring School and Workshop "Information and Uncertainty"
http://mcin.upol.cz/SSWIU-2014/

esf european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT