Data Dependencies in the Presence of Difference

Shaoxu Song

Tsinghua University
sxsong@tsinghua.edu.cn

Outline

Shaoxu Song Tsinghua University sxsong@tsinghua.edu.cn

Motivation

Data Dependencies traditionally for quality of Schema:
schema design, integrity constraints, query optimization, etc.

Data Dependencies recently for quality of Data:
data cleaning, data repairing, record matching, etc.

Table: Example instance of Empolyee

	name	institute	title	salary	ssn
t_1	John Depp	Tech. Univ.	Professor	60	111
t_2	J. Depp	Technical Univ.	Professor	60	111
t_3	J.C. Depp	Tech. University	Prof.	3	111
t_4	R. Depp	Western Univ.	Lecturer	30	222

Motivation

Identification function in schema-oriented issues,

- in conventional dependencies, e.g., FDs
- title $\rightarrow$ salary
- t_1[title] : *Professor* = t_2[title] : *Professor*
- t_1[salary] : 60 = t_2[salary] : 60

Difference semantics in data-oriented practice,

- on numerical values or text values, e.g., similar or dissimilar.
- title : *Professor* $\approx$ *Prof* .
- salary: 60k v.s. 3k

Shaoxu Song Tsinghua University sxsong@tsinghua.edu.cn

Differential Dependencies: Syntax

We propose a novel type of dependencies

- *differential dependencies* (DDs)
- in the form of $\phi_L[X] \rightarrow \phi_R[Y]$
- $\phi_L[X]$ and $\phi_R[Y]$ are differential functions, which specify distance constraints on attributes X and Y of R, respectively.

Constraints on difference

- for any two tuples (t_1, t_2) from an instance of R
- if their value differences (measured by certain distance metric) on attributes X agree with the differential function $\phi_L[X]$, $(t_1, t_2) \asymp \phi_L[X]$
- then their value differences on Y should also agree with the differential function $\phi_R[Y]$, $(t_1, t_2) \asymp \phi_R[Y]$

Example

A DD in a credit card transaction database can be

- DD_1 $[\text{cardno}(= 0) \wedge \text{position}(\geq 60)] \rightarrow [\text{transtime}(\geq 20)]$
- cardno$(= 0)$ states that two transactions have the same credit card no (the difference on attribute cardno is 0)
- position(≥ 60), transtime(≥ 20) are differential functions specified on attribute position, transtime, respectively

Constraints on difference

- If the distance of two transaction positions of a same cardno is ≥ 60 km (e.g., two different cities)
- they are probably two transactions happening at different time
- the difference between transtime should be ≥ 20 mins.

If two card transactions do not satisfy DD_1, one of the transactions could be a fraud.

Example

A DD in a price database of a flight, in decision support systems

- DD_2 $[\text{date}(\leq 7)] \rightarrow [\text{price}(\leq 100)]$
- states that the price difference of any two days in a week length should be less than 100 \$

Instead of a week length, another DD may specify

- DD_3 $[\text{date}(> 7, \leq 30)] \rightarrow [\text{price}(> 100, \leq 900)]$
- the price difference constraint of two days not in a week length but in a month length

Both DD_2 and DD_3 specify

- on the same embedded attributes date $\rightarrow$ price
- but with different constraint semantics, i.e., week and month.

Shaoxu Song | Tsinghua University sxsong@tsinghua.edu.cn

Related Work

Conditional functional dependencies (CFDs)

- $(X \rightarrow A, t_p)$
- make the FDs, originally hold for the whole table, valid only for a set of tuples specified by the conditions
- $([\text{country}, \text{zip}] \rightarrow [\text{street}], < \text{Finland}, \_ \parallel \_ >)$

Metric functional dependencies (MFDs)

- $X \xrightarrow{\delta} A$
- similarity metrics in the right-hand-side, for violation detection
- $\text{name} \xrightarrow{2} \text{address}$

Matching dependencies (MDs)

- $[X \approx] \rightarrow [A \rightleftharpoons]$
- "similar" semantics in the left-hand-side, for record matching
- $[\text{name} \approx] \wedge [\text{addr} \approx] \rightarrow [\text{tel} \rightleftharpoons]$

Comparison

CFDs introduce condition extension, which is still on identification semantics.

MFDs, MDs consider the "similar" semantics, on either determinant attributes X or dependent attributes Y

Our differential dependencies DDs

- $\phi_L[X] \rightarrow \phi_R[Y]$
- address more general difference constraints with various semantics
  - "similar" (e.g., price(≤ 100) in DD_2)
  - "dissimilar/different" (e.g., transtime(≥ 20) in DD_1),
  - or even more complicated ones (e.g., date($> 7, \leq 30$) in DD_3)
- allow setting difference constraints on both determinant attributes X and dependent attributes Y

Outline

Example: Violation Detection

To find the tuples that violate dependencies

- according to DD_2 $[\text{date}(\leq 7)] \rightarrow [\text{price}(\leq 100)]$
- t_3, t_4 are detected as violations to DD_2

FDs cannot express such constraints on difference

- t_3, t_4 cannot be detected by a FD date $\rightarrow$ price
- t_1, t_2 are detected as violations to FD by mistake

Table: Example of a price database

Tuple	Date	Price
t_1	2010.06.01	1,000
t_2	2010.06.01	1,050
t_3	2010.08.02	2,000
t_4	2010.08.03	3,000

Evaluation: Violation Detection

DDs compared with FDs with identification functions

- differential functions in the right-hand-side Y
  - detect violations more accurately
  - the detection precision is higher than FDs
- differential functions in the left-hand-side X
  - address more tuples with violations
  - the detection recall by using DDs is higher than FDs

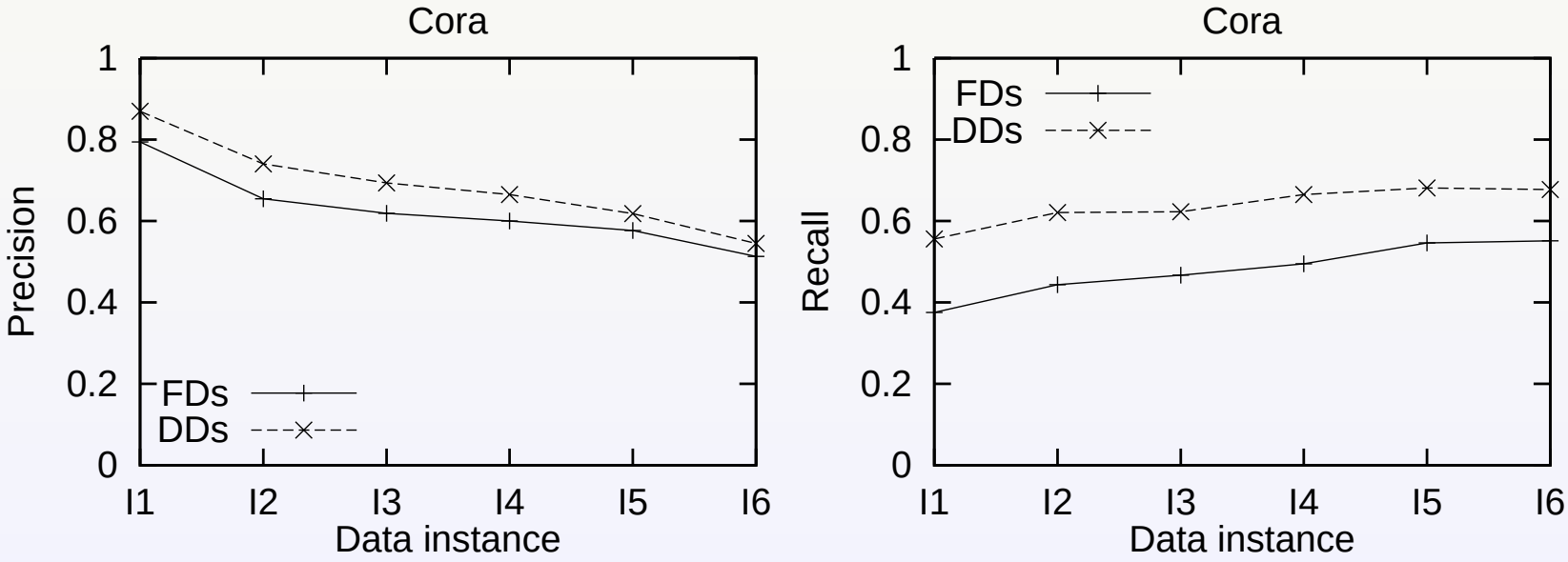


Figure: Violation detection accuracy

Example: Data Partition

To optimize data partition queries

- Integrity constraints (e.g., FDs or candidate keys) can be utilized to optimize the evaluation of queries
- known as the semantic query optimization

Consider a group-by query on distance conditions

```
SELECT * FROM Employee
GROUP BY
```
$\text{institute}(\leq 5) \wedge \text{title}(\leq 6)$

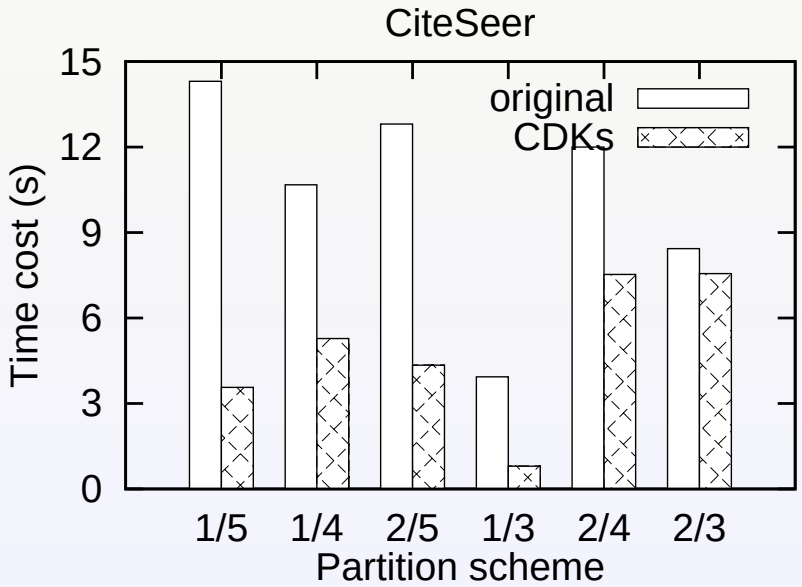
- according to $[\text{institute}(\leq 5)] \rightarrow [\text{institute}(\leq 5) \wedge \text{title}(\leq 6)]$
- rewrite the query by using $\text{institute}(\leq 5)$ only

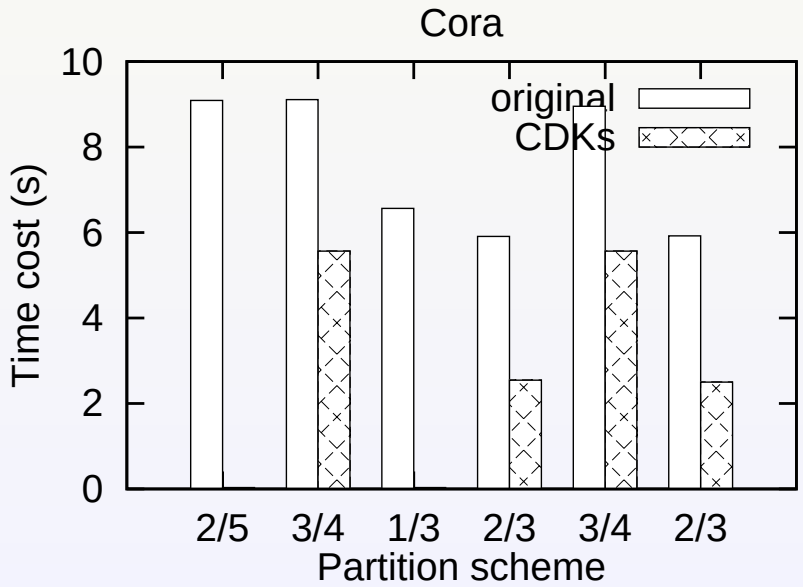
```
SELECT * FROM Employee
GROUP BY
```
$\text{institute}(\leq 5)$

Evaluation: Data Partition

Using *candidate differential key dependencies*, CDK dependencies

- In x-axis, each element a/b corresponds to a pair of reduced/original differential functions for partitioning queries
  - a denotes the cardinality of CDK
  - b denotes the cardinality of original partition scheme
- the smaller the rate a/b is, the more the performance can be improved





Example: Record Linkage

To identify duplicate record, a.k.a. record matching, merge-purge

- use DDs as matching rules
  - DD_1 $[\text{name}(\leq 5) \wedge \text{institute}(\leq 7)] \rightarrow [\text{ssn}(= 0)]$
  - t_1, t_2, whose name distance is ≤ 5, and institute distance is ≤ 7, probably denote the same employee with identical ssn
- another valid matching rule on same attributes
  - DD_2 $[\text{name}(\leq 3) \wedge \text{institute}(\leq 15)] \rightarrow [\text{ssn}(= 0)]$
  - t_2, t_3 detected as duplicates by DD_2, not detected by DD_1

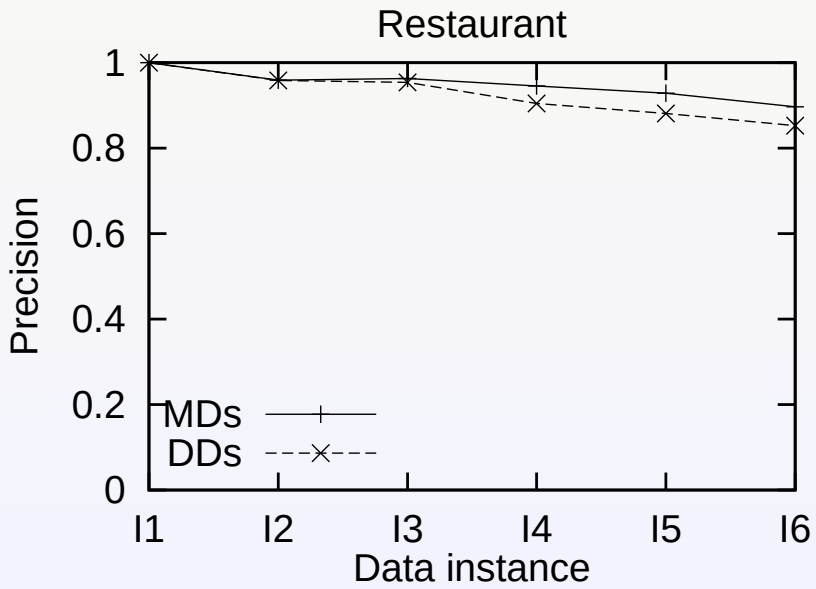
Table: Example instance of Empolyee

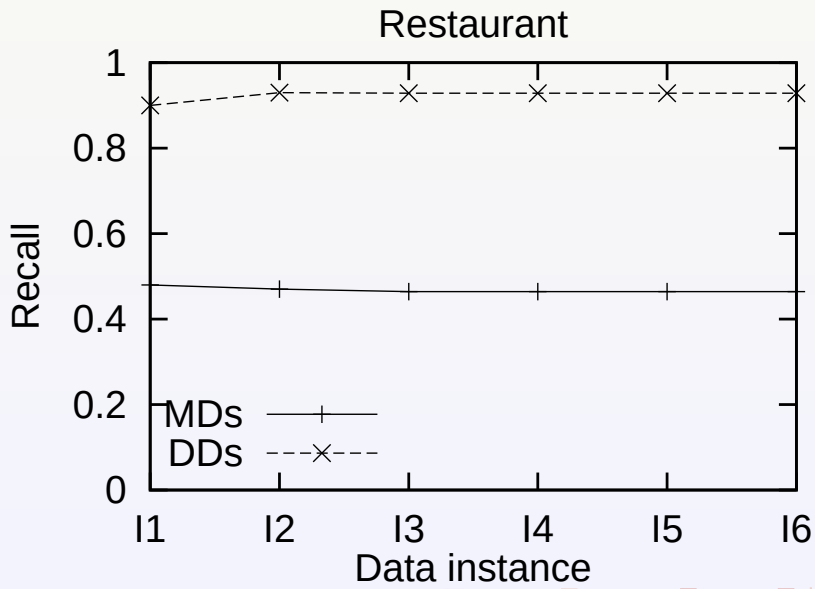
	name	institute	title	salary	ssn
t_1	John Depp	Tech. Univ.	Professor	60	111
t_2	J. Depp	Technical Univ.	Professor	60.2	111
t_3	J.C. Depp	Tech. University	Prof.	30	111
t_4	R. Depp	Western Univ.	Lecturer	30	222

Evaluation: Record Linkage

DDs compared MDs,

- MDs associate only one differential function on each attribute
- DDs can specify various differential functions on one attribute
- DDs address more matching rules
- recall of DDs is significantly higher
- DDs have comparable precision as MDs, both are valid matching rules





Outline

Differential Function: Intersection

The *intersection* of $\phi_1[Z]$ and $\phi_2[Z]$ on the same attributes Z is $\phi_3[Z] = \phi_1[Z] \wedge \phi_2[Z]$

- any $(t_1, t_2) \asymp \phi_1[Z]$ and $(t_1, t_2) \asymp \phi_2[Z]$, then $(t_1, t_2) \asymp \phi_3[Z]$
- any $(t_1, t_2) \not\asymp \phi_1[Z]$ or $(t_1, t_2) \not\asymp \phi_2[Z]$, then $(t_1, t_2) \not\asymp \phi_3[Z]$
- $[\text{name}(\leq 9)] \wedge [\text{name}(\leq 7)] = [\text{name}(\leq 7)]$

Apply intersection between $\phi_1[X]$ and $\phi_2[Y]$ on different attributes X and Y

- Let $Z = X \cap Y$

$$\begin{aligned}\phi_1[X] \wedge \phi_2[Y] =& (\phi_1[X \setminus Z] \wedge \phi_1[Z]) \wedge (\phi_2[Z] \wedge \phi_2[Y \setminus Z]) \\ =& \phi_1[X \setminus Z] \wedge (\phi_1[Z] \wedge \phi_2[Z]) \wedge \phi_2[Y \setminus Z].\end{aligned}$$

- $[\text{name}(\leq 5) \wedge \text{address}(\leq 12)] \wedge [\text{address}(\leq 10)] =$ $[\text{name}(\leq 5) \wedge \text{address}(\leq 10)]$

Differential Function: Subsumption

Intuitively, the semantics of "similar" subsumes identification

- any two values that are "identical" (with distance $= 0$)
- can always be interpreted as "similar" (with distance ≤ 9)

Definition

Let $\phi_1[Z]$ and $\phi_2[Z]$ be two differential functions on attributes Z

- If any tuple pair $(t_1, t_2) \asymp \phi_2[Z]$ always agree $(t_1, t_2) \asymp \phi_1[Z]$
- we say that $\phi_1[Z]$ *subsumes* $\phi_2[Z]$, written $\phi_1[Z] \succeq \phi_2[Z]$

For example

- $\phi_1[\text{name}] = [\text{name}(\leq 9)]$ subsumes $\phi_2[\text{name}] = [\text{name}(\leq 7)]$
  - denoted by $[\text{name}(\leq 9)] \succeq [\text{name}(\leq 7)]$
  - a distance value of name that agrees ≤ 7 will always agree ≤ 9
- $[\text{date}(\leq 30)] \succeq [\text{date}(> 7, \leq 30)]$; $[\text{addr}(\leq 9)] \succeq [\text{addr}(= 0)]$

Differential Dependency

Consider an instance I of relation R

- $(t_1, t_2) \asymp \phi_L[X]$ denotes tuples (t_1, t_2) having distance agreeing $\phi_L[X]$
- I *satisfies* a DD, $I \vDash \phi_L[X] \rightarrow \phi_R[Y]$,
  if any two tuples t_1 and t_2 in I having metric distances $(t_1, t_2) \asymp \phi_L[X]$ must agree $(t_1, t_2) \asymp \phi_R[Y]$
- I satisfies a set Σ of DDs, $I \vDash \Sigma$
  if $I \vDash \phi_L[X] \rightarrow \phi_R[Y]$ for each $\phi_L[X] \rightarrow \phi_R[Y] \in \Sigma$.

Proposition

For two differential functions $\phi_L[X]$ and $\phi_R[Y]$, if $Y \subseteq X$ and $\phi_R[Y] \succeq \phi_L[Y]$, then $\phi_L[X] \rightarrow \phi_R[Y]$.

- a trivial DD, always holds
- $[\text{name}(\leq 5) \wedge \text{address}(\leq 10)] \rightarrow [\text{address}(\leq 12)]$

Logical Implication

Example

Consider two DDs,

$$\text{DD}_4 \quad [\text{name}(\leq 7)] \rightarrow [\text{address}(\leq 1)],$$

$$\text{DD}_5 \quad [\text{address}(\leq 5)] \rightarrow [\text{salary}(\leq 50)].$$

- any two tuples t_1 and t_2 having name distance ≤ 7,
- according to DD_4, their distance on address should be ≤ 1,
- (t_1, t_2) agree address(≤ 5) as well.
- the salary distance of t_1 and t_2 should be ≤ 50 according to DD_5

We can imply another DD,

$$\text{DD}_6 \quad [\text{name}(\leq 7)] \rightarrow [\text{salary}(\leq 50)].$$

Implication Problem

Let Σ_1 and Σ_2 be two sets of DDs.

- Σ_1 *logically implies* Σ_2, $\Sigma_1 \models \Sigma_2$
  if for all relation instance I, $I \models \Sigma_1$ implies $I \models \Sigma_2$
- Σ_1 and Σ_2 are *equivalent*, $\Sigma_1 \equiv \Sigma_2$
  if $\Sigma_1 \models \Sigma_2$ and $\Sigma_2 \models \Sigma_1$

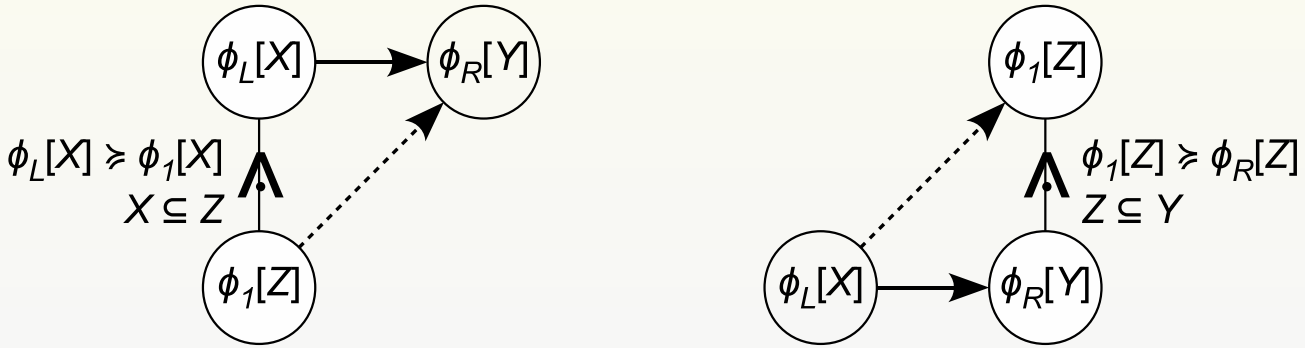
The *implication problem*

- given a consistent set Σ of DDs and another DD $\phi_L[X] \rightarrow \phi_R[Y]$
- to decide whether Σ can imply this DD, $\Sigma \models \phi_L[X] \rightarrow \phi_R[Y]$
- For example, $\{\mathrm{DD}_4, \mathrm{DD}_5\} \models \mathrm{DD}_6$

Implication based-on Subsumption

Given a DD $\phi_L[X] \rightarrow \phi_R[Y]$

- $\phi_1[Z] \rightarrow \phi_R[Y]$ can be implied, if $X \subseteq Z, \phi_L[X] \succeq \phi_1[X]$
- $\phi_L[X] \rightarrow \phi_1[Z]$ can be implied, if $Z \subseteq Y, \phi_1[Z] \succeq \phi_R[Z]$



For example, consider a DD $[\text{name}(\leq 7)] \rightarrow [\text{address}(\leq 1)]$, it implies

- $[\text{name}(\leq 5)] \rightarrow [\text{address}(\leq 1)]$
- $[\text{name}(\leq 7)] \rightarrow [\text{address}(\leq 2)]$

Differential Key

Key: $t_1[R] = t_2[R]$ according to $t_1[K] = t_2[K]$ on a key $K \subseteq R$

A *differential key* $\phi_2[K]$ relative to $\phi_1[R]$

- is a differential function that can determine $\phi_1[R]$
- a *differential key dependency* $\phi_2[K] \rightarrow \phi_1[R]$ with $K \subseteq R$ and $\phi_2[K] \succeq \phi_1[K]$

For example,

- [position(≥ 20)] is a differential key relative to [position(≥ 20) $\wedge$ area(≥ 5)]
- according to the following differential key dependency, [position(≥ 20)] $\rightarrow$ [position(≥ 20) $\wedge$ area(≥ 5)]

Candidate Differential Key

A naïve key relative to $\phi_1[R]$ is $\phi_1[R]$ itself

A *candidate differential key* (CDK) $\phi_c[K]$ is

- an *irreducible* differential key relative to $\phi_1[R]$,
- there does not exist any $\phi_2[L]$ such that $L \subseteq K$, $\phi_2[L] \succeq \phi_c[L]$ and $\phi_2[L] \rightarrow \phi_1[R]$.

A CDK

- not only has a minimal cardinality as candidate keys on FDs,
- but also should be the one not subsumed by others.

CDKs are useful in applications like data partition

Outline

Discovery Problem

Discovery from data

- given a relation instance I
- discover candidate differential keys and minimal cover of differential dependencies that hold in I

The hardness

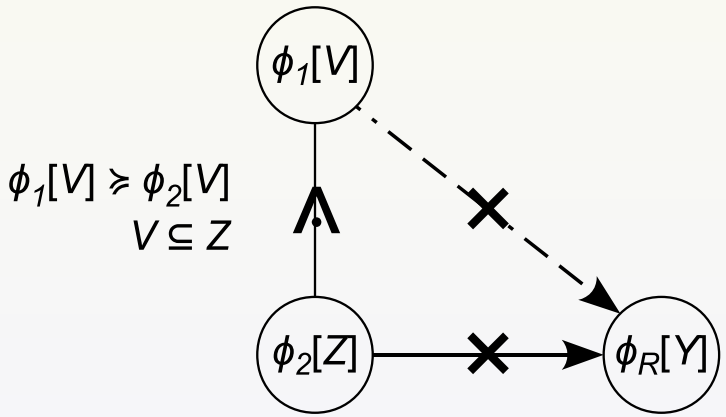
- a minimal cover of FDs, that hold in a relation instance I, can be exponentially large in the number of attributes
- FDs are considered as special cases of DDs where all the differential constraints are set to $= 0$
- DDs subsume FDs, could be exponentially large as well

Negative Pruning

Motivation: pruning candidates of DDs, in order to avoid evaluating all possible $\phi_L[X] \rightarrow \phi_R[Y]$ in I

Lemma

For any $\phi_1[V], \phi_2[Z]$ *having* $V \subseteq Z, \phi_1[V] \succeq \phi_2[V]$, *if* $I \nvDash \phi_2[Z] \rightarrow \phi_R[Y]$, *then* $I \nvDash \phi_1[V] \rightarrow \phi_R[Y]$



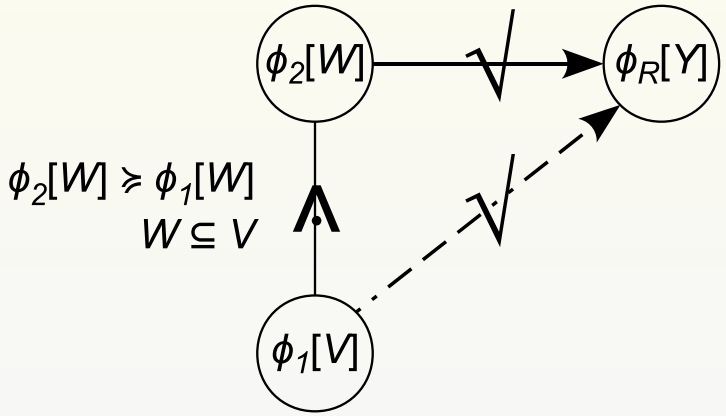
Example: if $[\text{name}(\leq 5)] \rightarrow \phi_R[Y]$ not hold in I, then $[\text{name}(\leq 7)] \rightarrow \phi_R[Y]$ not hold either without evaluation in I

Worst case: all the candidates hold in the given instance I

Positive Pruning

Lemma

For any $\phi_1[V], \phi_2[W]$ *having* $W \subseteq V, \phi_2[W] \succeq \phi_1[W]$, *if* $I \vDash \phi_2[W] \rightarrow \phi_R[Y]$, *then* $I \vDash \phi_1[V] \rightarrow \phi_R[Y]$



Example: if $[\text{name}(\leq 7)] \rightarrow \phi_R[Y]$ holds in I, then $[\text{name}(\leq 5)] \rightarrow \phi_R[Y]$ must hold without evaluation in I

Worst case: all the candidates do not hold in the given instance I

Hybrid approach with both positive and negative pruning, used by turns.

Instance Exclusion

Motivation: avoiding evaluating the entire I.

- one differential function subsumes another
- the set of tuples agreeing on the former one should be a super set of the latter one

Considers all the pairs of tuples in I.

$$D(I) = \{(t_i, t_j) \mid \forall t_i, t_j \in I\}.$$

Given any DD $\phi_L[X] \rightarrow \phi_R[Y]$, we define $D(I, \phi_L[X], \neg\phi_R[Y]) =$

$$\{(t_i, t_j) \in D(I) \mid (t_i, t_j) \asymp \phi_L[X], (t_i, t_j) \not\asymp \phi_R[Y]\},$$

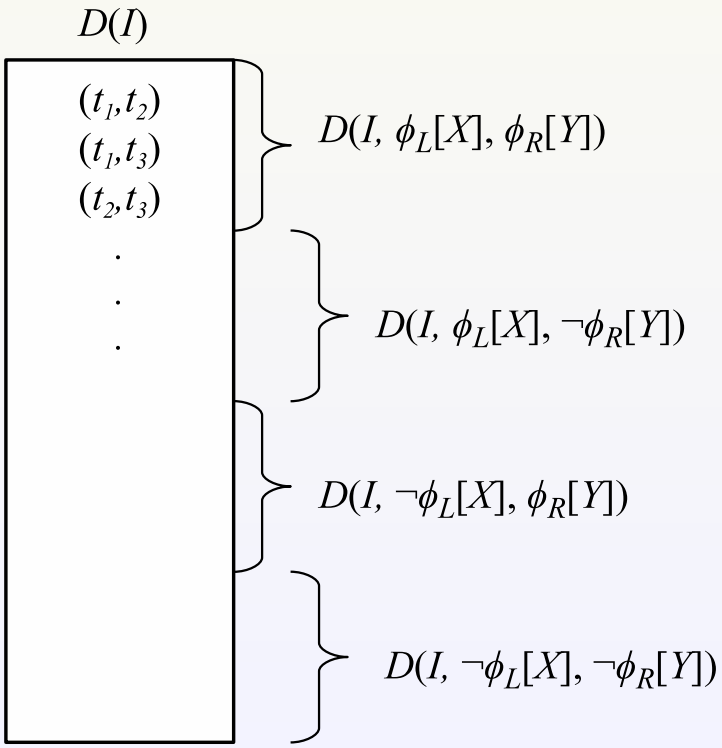
that is, the tuple pairs agreeing $\phi_L[X]$ but not agreeing $\phi_R[Y]$.

Instance Exclusion

Lemma

An instance I satisfies a DD, $I \vDash \phi_L[X] \rightarrow \phi_R[Y]$, *iff* $D(I, \phi_L[X], \neg\phi_R[Y]) = \emptyset$.

During the discovery, for a candidate $\phi_L[X] \rightarrow \phi_R[Y]$, have to evaluate whether $D(I, \phi_L[X], \neg\phi_R[Y]) = \emptyset$.



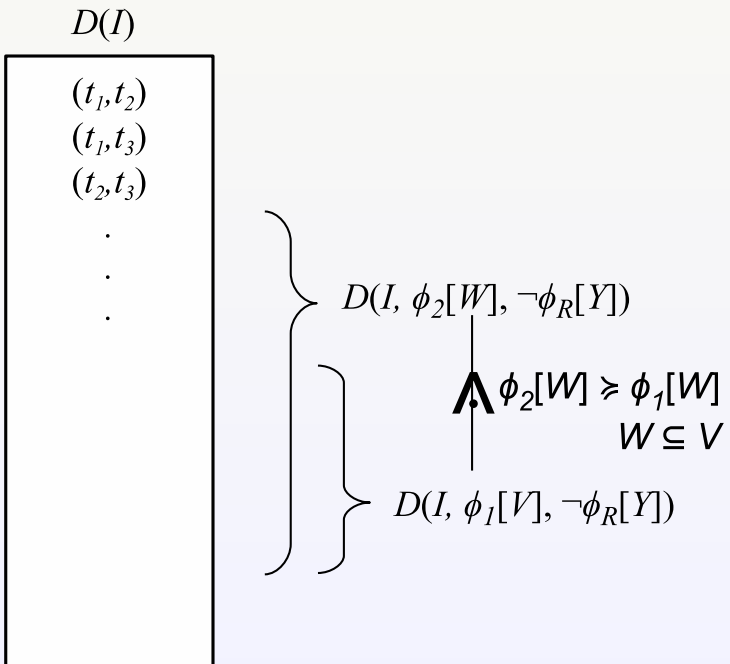
Instance Exclusion

Lemma

For any $\phi_1[V], \phi_2[W]$ *having* $W \subseteq V, \phi_2[W] \succeq \phi_1[W]$, *we have* $D(I, \phi_1[V], \neg\phi_R[Y]) \subseteq D(I, \phi_2[W], \neg\phi_R[Y])$.

Suppose that a current $D(I, \phi_2[W], \neg\phi_R[Y]) \neq \emptyset$

- instead of considering the entire $D(I)$
- use $D(I, \phi_2[W], \neg\phi_R[Y])$ to compute $D(I, \phi_1[V], \neg\phi_R[Y])$



Experiments

Evaluate the time performance of discovery approaches

- scale well with the increase of tuples in an instance I
- $O(n^2)$ with respect to the number of tuples n in the instance I
- instance exclusion performs well

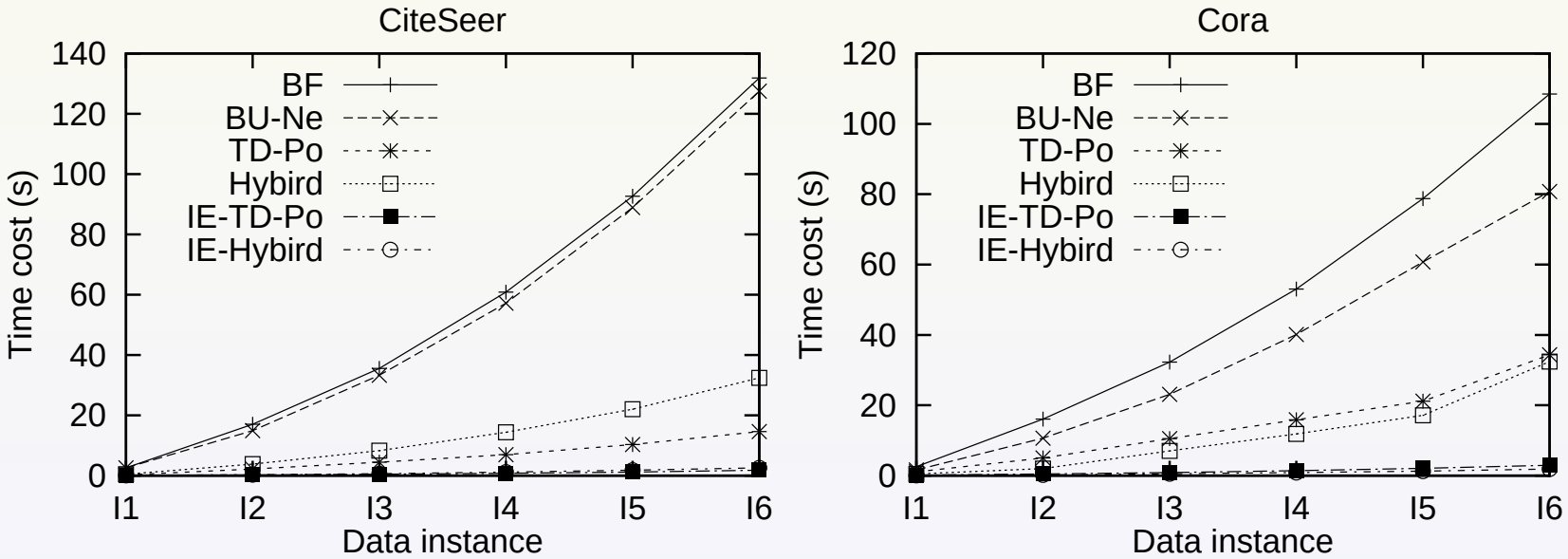


Figure: DDs discovery performance on various instance I

Experiments

- discovery cost increases exponentially in the number of attributes in a schema
- can achieve several orders of magnitude improvement compared with brute-force one

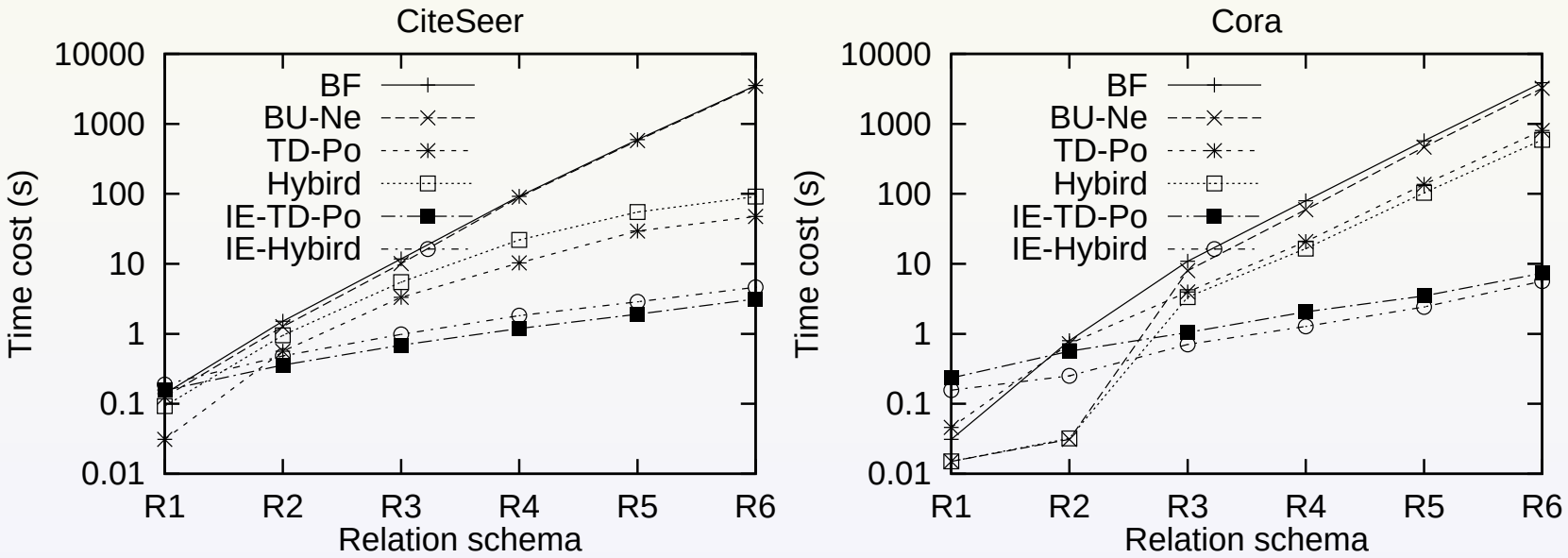


Figure: DDs discovery performance on various schema R

Outline

Conclusions

We propose a novel class of dependencies, differential dependencies (DDs), which specify constraints on distance.

Theory

- formal definitions of DDs and differential keys
- subsumption order relation of differential functions
- reasoning about DDs
  - consistency of DDs, NP-complete
  - implication of DDs, co-NP-complete
  - closure of a differential function
  - a sound and complete inference system, proof
  - minimal cover for DDs

Practice

- discovery of DDs and differential keys from data.
- application of DDs and differential keys.

Future Work

Approximate differential dependencies

- "almost" hold in a data instance
- evaluation measure, efficient computation
  - Implication of approximate differential dependencies
  - Hardness analysis of computing error measure
  - Approximation algorithms computing error measure
  - Experiments of approximation validation

Further extensions

- data repairing with DDs
- conditioning DDs in a subset of tuples
- integrity rules in dataspaces

Data Dependencies in the Presence of Difference

Shaoxu Song

Tsinghua University
sxsong@tsinghua.edu.cn

Closure

The *closure* of $\phi_L[X]$ under Σ, $(\phi_L[X])^+$

- is also a differential function
- the intersection of the set of differential functions that can be determined by $\phi_L[X]$ according to DDs in Σ

$$(\phi_L[X])^+ = \bigwedge\{\phi_R[Y] \mid \Sigma \vDash \phi_L[X] \rightarrow \phi_R[Y]\}$$

- the closure of [name(≤ 7)] under $\{\mathrm{DD}_4, \mathrm{DD}_5\}$ is [name(≤ 7) $\wedge$ address(≤ 1) $\wedge$ salary(≤ 50)]

It is natural that $\phi_L[X] \rightarrow (\phi_L[X])^+$.

Closure

To imply a DD is essentially to compute the corresponding closure $(\phi_L[X])^+$ of $\phi_L[X]$

Lemma

Let Σ be a set of DDs and $\phi_1[Z] = (\phi_L[X])^+$ be the closure of $\phi_L[X]$ with respect to Σ.

- *Consider a DD $\phi_L[X] \rightarrow \phi_R[Y]$,*
- *$\Sigma \vDash \phi_L[X] \rightarrow \phi_R[Y]$ iff $Y \subseteq Z$ and $\phi_R[Y] \succeq \phi_1[Y]$.*

For example,

- [salary(≤ 50)] subsumes the projection on salary of the closure of [name(≤ 7)] under $\{\mathrm{DD}_4, \mathrm{DD}_5\}$
- it implies DD_6 [name(≤ 7)] $\rightarrow$ [salary(≤ 50)]

Inference System

A1. If $Y \subseteq X$ and $\phi_L[Y] = \phi_R[Y]$, then $\Sigma \vdash_{\mathcal{I}} \phi_L[X] \rightarrow \phi_R[Y]$.

A2. If $\Sigma \vdash_{\mathcal{I}} \phi_L[X] \rightarrow \phi_R[Y]$, then
$\Sigma \vdash_{\mathcal{I}} \phi_L[X] \wedge \phi_1[Z] \rightarrow \phi_R[Y] \wedge \phi_1[Z]$.

A3. If $\Sigma \vdash_{\mathcal{I}} \phi_L[X] \rightarrow \phi_1[Z]$, $\phi_1[Z] \preceq \phi_2[Z]$ and
$\Sigma \vdash_{\mathcal{I}} \phi_2[Z] \rightarrow \phi_R[Y]$, then $\Sigma \vdash_{\mathcal{I}} \phi_L[X] \rightarrow \phi_R[Y]$.

A4. If $\Sigma \vdash_{\mathcal{I}} \phi_L[X] \wedge \phi_i[B] \rightarrow \phi_R[Y], 1 \leq i \leq k$, and
$(\Sigma, \overline{\phi_1[B]} \wedge \cdots \wedge \overline{\phi_k[B]})$ is inconsistent, then
$\Sigma \vdash_{\mathcal{I}} \phi_L[X] \rightarrow \phi_R[Y]$.

Theorem

The set $\mathcal{I}$ of inference rules is

- *(sound), if $\Sigma \vdash_{\mathcal{I}} \phi_L[X] \rightarrow \phi_R[Y]$ then $\Sigma \vDash \phi_L[X] \rightarrow \phi_R[Y]$,*
- *(complete), if $\Sigma \vDash \phi_L[X] \rightarrow \phi_R[Y]$ then*
  $\Sigma \vdash_{\mathcal{I}} \phi_L[X] \rightarrow \phi_R[Y]$,

for logical implication of DDs.

Example: Inference

Example

We consider a set Σ of DDs as follows:

$$\begin{aligned} \text{DD}_7 \quad & [\text{d}(\geq 1, \leq 7) \wedge \text{p}(< 10)] \rightarrow [\text{a}(\leq 150)], \\ \text{DD}_8 \quad & [\text{p}(\geq 10)] \rightarrow [\text{a}(\leq 100)]. \end{aligned}$$

Let DD_9 be another DD

$$\text{DD}_9 \quad [\text{d}(\geq 1, \leq 7)] \rightarrow [\text{a}(\leq 150)].$$

We show that $\Sigma \vdash_{\mathcal{I}} \text{DD}_9$ can be proved by the following steps.

1. $[\text{d}(\geq 1, \leq 7) \wedge \text{p}(\geq 10)] \rightarrow [\text{d}(\geq 1, \leq 7) \wedge \text{a}(\leq 100)]$ by *A2*, DD_8
2. $[\text{d}(\geq 1, \leq 7) \wedge \text{a}(\leq 150)] \rightarrow [\text{a}(\leq 150)]$ by *A1*
3. $[\text{d}(\geq 1, \leq 7) \wedge \text{p}(\geq 10)] \rightarrow [\text{a}(\leq 150)]$ by *A3*, 1. 2.
4. $[\text{d}(\geq 1, \leq 7)] \rightarrow [\text{a}(\leq 150)]$ by *A4*, 3. DD_7

Minimal Cover

A *minimal cover* Σ_c for Σ is a set of DDs such that Σ_c

- is logically equivalent to Σ, i.e., $\Sigma_c \equiv \Sigma$
- is *minimal* according to the following properties:

C1. (left-reduced), for any $\phi_L[X] \rightarrow \phi_R[Y] \in \Sigma_c$, there does not exist any $\phi_1[W]$ such that $W \subseteq X$, $\phi_1[W] \succeq \phi_L[W]$ and $\Sigma_c \vDash \phi_1[W] \rightarrow \phi_R[Y]$.

C2. (right-subsumed), for any $\phi_L[X] \rightarrow \phi_R[Y] \in \Sigma_c$, there does not exist any $\phi_1[W]$ such that $Y \subseteq W$, $\phi_1[Y] \preceq \phi_R[Y]$ and $\Sigma_c \vDash \phi_L[X] \rightarrow \phi_1[W]$.

C3. (non-redundant), there does not exist a cover Σ' of Σ such that $\Sigma' \subset \Sigma_c$.

Example: Minimal Cover

Consider $\Sigma = \{\mathrm{DD}_4, \mathrm{DD}_5, \mathrm{DD}_6\}$ in Example 2

- a minimal cover can be $\Sigma_c = \{\mathrm{DD}_4, \mathrm{DD}_5\}$
- Σ_c can imply DD_6
- by removing DD_4 or DD_5 from Σ_c, it is no longer a cover of Σ

Consider $\Sigma = \{\mathrm{DD}_7, \mathrm{DD}_8, \mathrm{DD}_9\}$ in Example 9

- a minimal cover can be $\Sigma_c = \{\mathrm{DD}_8, \mathrm{DD}_9\}$
- $\Sigma' = \{\mathrm{DD}_7, \mathrm{DD}_8\}$ is not a minimal cover
- since DD_7 is not left-reduced and can be implied by DD_9 by augmentation rule *A2*.