Qualitative Data Cleaning

Xu Chu   Ihab Ilyas

University of Waterloo

Many Definitions and One Goal

"Extract Value from Data"

- For that we ..
  - Remove errors
  - Fill missing info
  - Transform units and formats
  - Map and align columns
  - Remove duplicate records
  - Fix integrity constraints violations

Many Definitions and One Goal

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

NYtimes August, 2014

Yes big data is a big business opportunity, but the business value won't be realized if the information isn't governed

Forbes Business

Many Technical Challenges

Record Linkage and Deduplication

- Similarity measures
- Machine learning for classifying pairs as duplicates or not (unsupervised, supervised, and active)
- Clustering and handling of transitivity
- Merging and consolidation of records

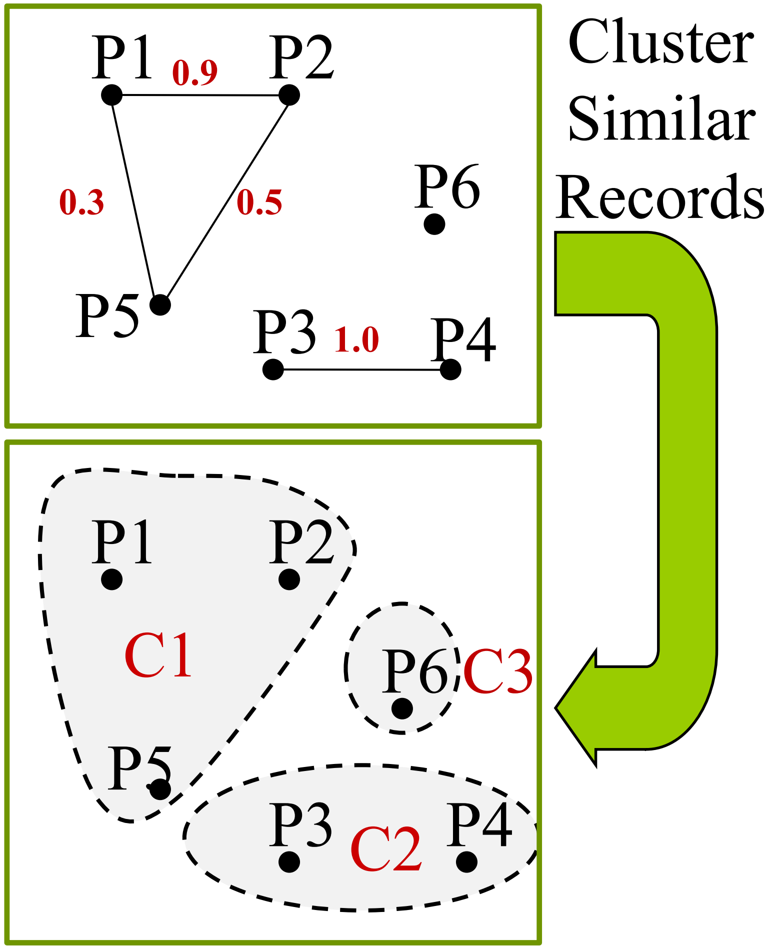
A major firm spends 6 months on a single deduplication project of a subset of their data sources

Example: Data Deduplication

Unclean Relation

ID	name	ZIP	Income
P1	Green	51519	30k
P2	Green	51518	32k
P3	Peter	30528	40k
P4	Peter	30528	40k
P5	Gree	51519	55k
P6	Chuck	51519	30k

Compute Pair-wise Similarity



Cluster Similar Records

Merge Clusters

Clean Relation

ID	name	ZIP	Income
C1	Green	51519	39k
C2	Peter	30528	40k
C3	Chuck	51519	30k

Many Technical Challenges

Missing Values

- Interpreting different types of **`Nulls`**
- Certain answer semantics on possible worlds (many.. many papers)
- Closed world vs. open-world assumptions and multiple interesting hardness results

Most real data collected from sensors, surveys, agents, have a high percentage of N/A or nulls, special values (99999) etc.

Many Technical Challenges

More Complex Integrity Constraints

- A declarative language to express data quality rules
- Ad-hoc repair algorithm to repair violations for each data quality formalism under certain minimality requirements
- Limited expressiveness (e.g., FD) to get tangible results

Unfortunately rarely expressed in practice. Most curation tools are rule-based implemented in imperative language

Example ICs

	ID	FN	LN	ROLE	CITY	ST	SAL
t_1	105	Anne	Nash	M	NYC	NY	110
t_2	211	Mark	White	E	SJ	CA	80
t_3	386	Mark	Lee	E	NYC	AZ	75
t_4	235	John	Smith	M	NYC	NY	1200

Employee Table

Functional dependency:

$$City \rightarrow ST$$

Example ICs

	ID	FN	LN	ROLE	CITY	ST	SAL
t_1	105	Anne	Nash	M	NYC	NY	110
t_2	211	Mark	White	E	SJ	CA	80
t_3	386	Mark	Lee	E	NYC	AZ	75
t_4	235	John	Smith	M	NYC	NY	1200

Employee Table

Business Rule:
Two employees of the same role, the one who lives in NYC cannot earn less than the one who does not live in NYC

Example ICs

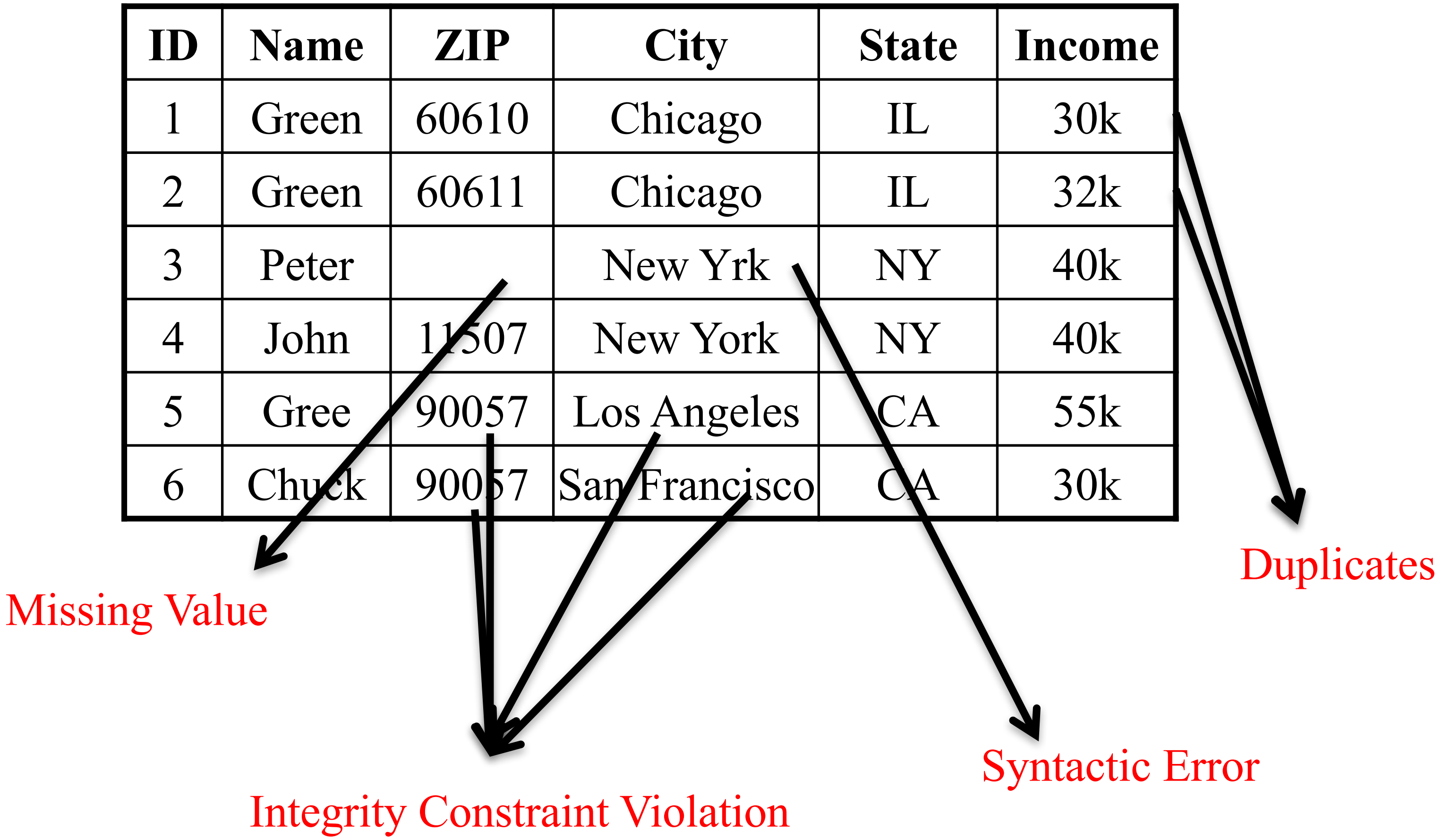
	ID	FN	LN	ROLE	CITY	ST	SAL
t_1	105	Anne	Nash	M	NYC	NY	110
t_2	211	Mark	White	E	SJ	CA	80
t_3	386	Mark	Lee	E	NYC	AZ	75
t_4	235	John	Smith	M	NYC	NY	1200

Employee Table

Business Rule:

Two employees of the same role in the same city, their salary difference cannot be greater than 100

Common Data Quality Issues



ID	Name	ZIP	City	State	Income
1	Green	60610	Chicago	IL	30k
2	Green	60611	Chicago	IL	32k
3	Peter		New Yrk	NY	40k
4	John	11507	New York	NY	40k
5	Gree	90057	Los Angeles	CA	55k
6	Chuck	90057	San Francisco	CA	30k

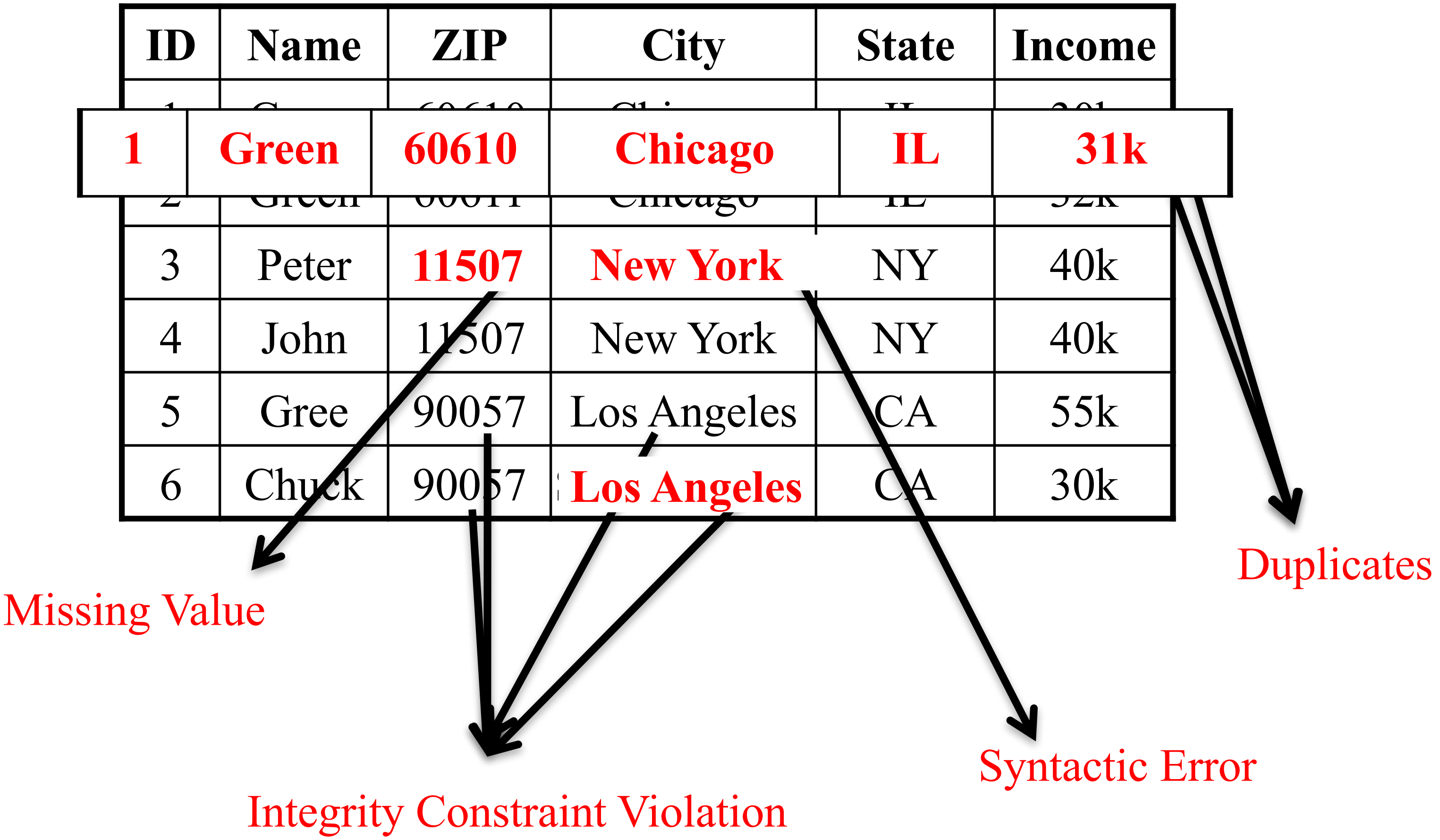
Missing Value

Integrity Constraint Violation

Syntactic Error

Duplicates

Common Data Quality Issues



ID	Name	ZIP	City	State	Income
1	Green	60610	Chicago	IL	31k
2	Green	60611	Chicago	IL	32k
3	Peter	11507	New York	NY	40k
4	John	11507	New York	NY	40k
5	Gree	90057	Los Angeles	CA	55k
6	Chuck	90057	Los Angeles	CA	30k

Missing Value

Integrity Constraint Violation

Syntactic Error

Duplicates

Data Cleaning Process

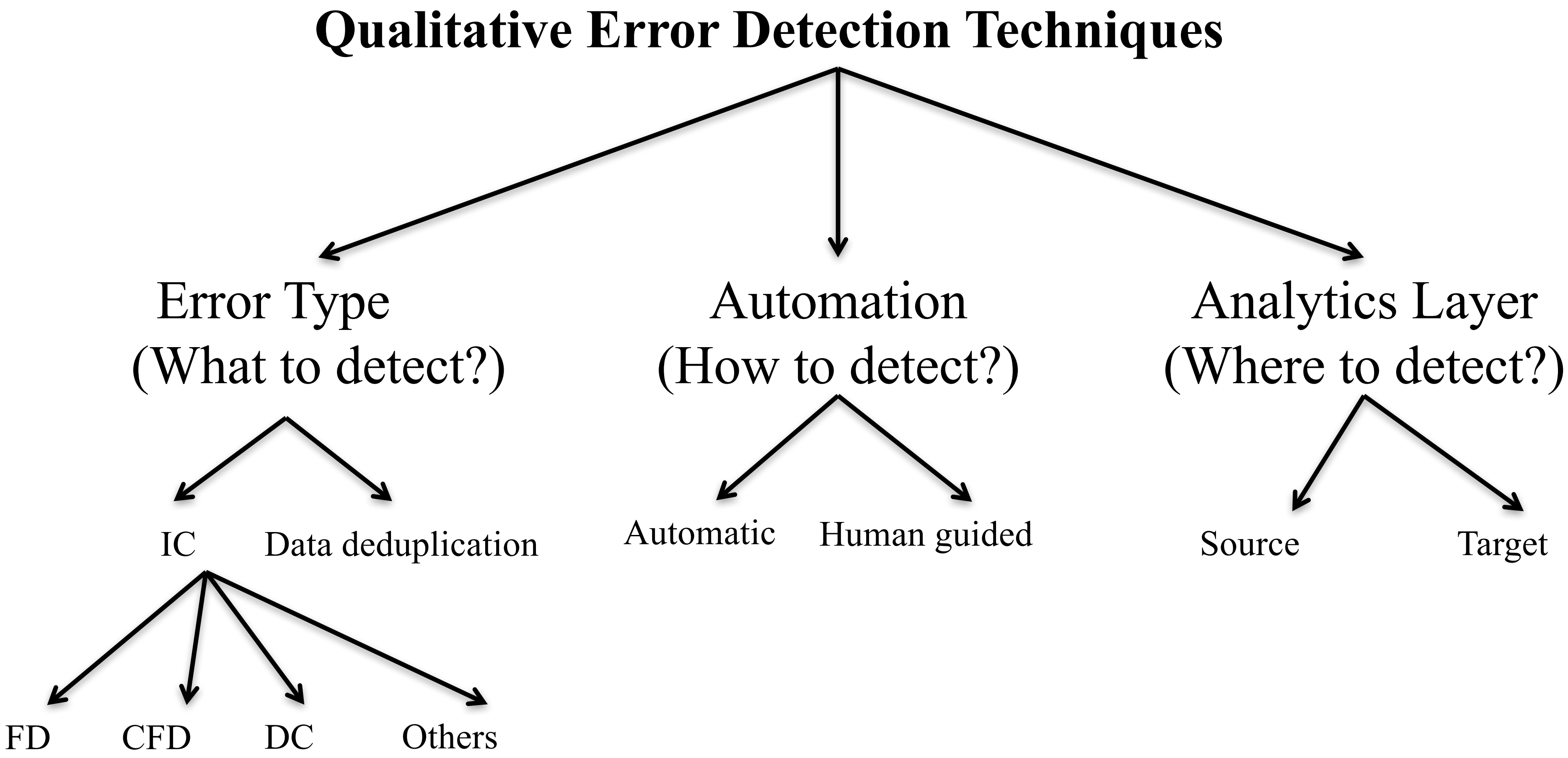
- Error Detection
  - Qualitative
  - Quantitative (outlier detection)

- Error Repairing
  - Transformation scripts
  - Human involvement

We Will Not Cover

- Details of Deduplication
  - Multiple surveys and tutorials
- Data Profiling: discovering FDs, INDs, etc.
  - Wenfei Fan and Floris Geerts synthesis lecture book
  - Ziawasch Abedjan et al. tutorial
- Consistent Query Answering
  - Leo Betrossi synthesis lecture book

Error Detection Techniques Taxonomy



Qualitative Error Detection Techniques

Error Type
(What to detect?)

Automation
(How to detect?)

Analytics Layer
(Where to detect?)

IC

Data deduplication

Automatic

Human guided

Source

Target

FD

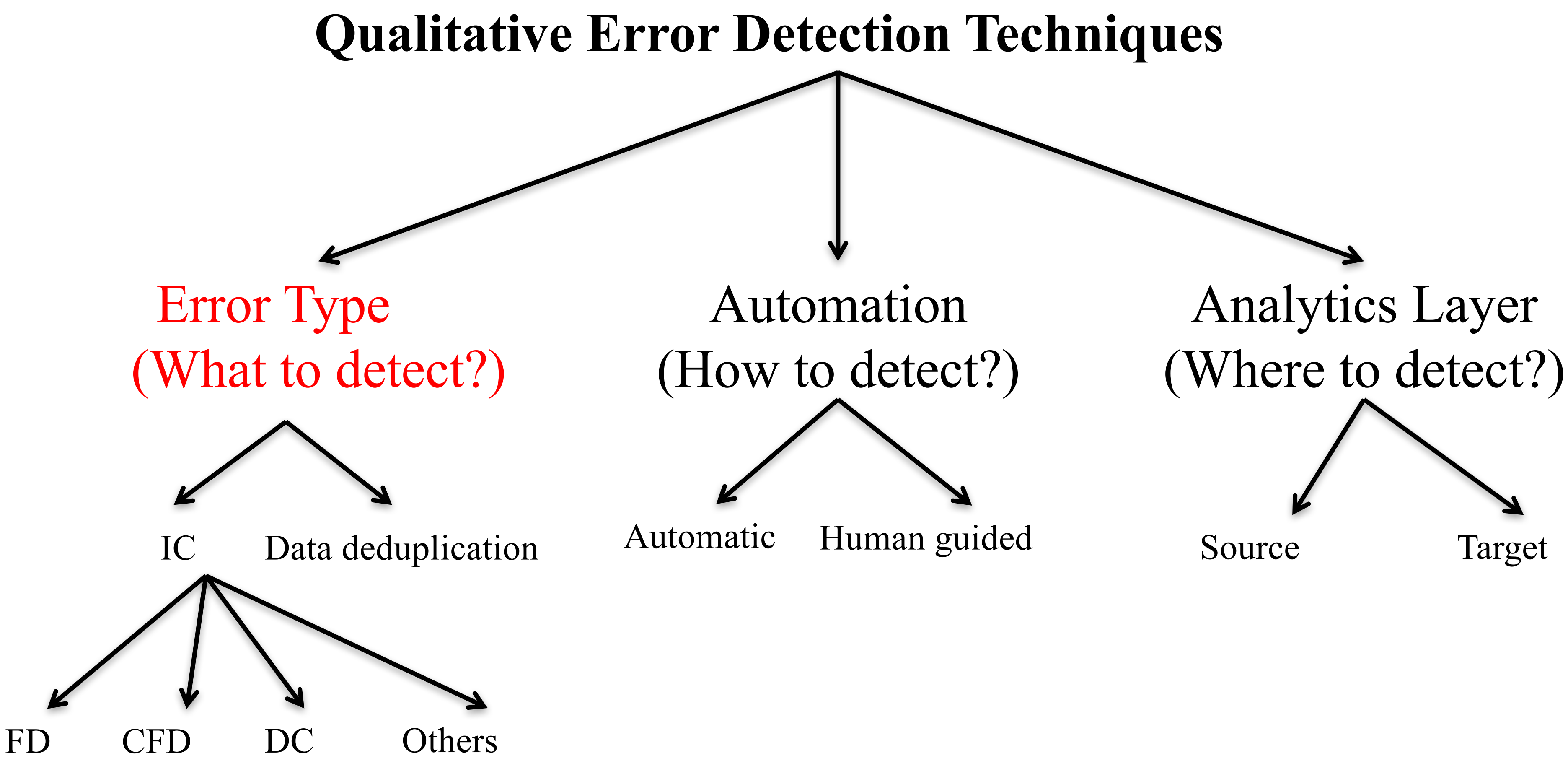
CFD

DC

Others

[Ilyas and Chu, Foundations and Trends in Database Systems, 2015]

Error Detection Techniques Taxonomy



Qualitative Error Detection Techniques

Error Type
(What to detect?)

IC

FD

CFD

DC

Others

Data deduplication

Automation
(How to detect?)

Automatic

Human guided

Analytics Layer
(Where to detect?)

Source

Target

FDs and CFDs [Bohannon et al, ICDE 2007]

- Functional Dependency (FD): X → Y
  - Example: City → ST or Name,Phone → ID
- Conditional Functional Dependency (CFD): $(X \rightarrow Y, T_p)$
  - An FD defined on a subset of the data
  - Example:
    - ZIP → Street is valid on subset of the data where Country = "England"
    - AC = 020 → City = London

Matching Dependencies (MDs) [Fan et al, VLDB 2009]

Tran

FN	LN	St	City	AC	Post	Phn	Item
Robert	Brady	5 Wren St	Ldn	020	WC1H 9SE	3887644	watch
Robert	Brady	Null	Ldn	020	WC1E 7HX	3887644	necklace

Master: Card

FN	LN	St	City	AC	Zip	Tel
Robert	Brady	5 Wren St	Ldn	020	WC1H 9SE	3887644

MD: Tran[LN, City, St, Post] = card[LN, City, St, Zip] ^
Tran[FN] ≈ Card[FN] → Tran[FN, Phn] ⇔ Card[FN, Tel]

[Fan et al, SIGMOD 2011]

Denial Constraints (DCs) [Chu et al, VLDB 2013]

Formal Definition:

$$\varphi : \forall t_\alpha, t_\beta, t_\gamma, \ldots \in R, \neg(P_1 \wedge \ldots \wedge P_m)$$

P_i: $t_x.A \ \theta \ t_y.B$ or $t_x.A \ \theta \ c$

$x, y \in \{\alpha, \beta, \ldots\}$, and $A, B \in R$, c is a constant

- A universal constraint dictates a set of predicate cannot be true together
- Each predicate express a relationship between two cells, or a cell and a constant

Denial Constraints (DCs)

Functional dependency:
$CITY \Rightarrow ST$

$$\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.CITY = t_\beta.CITY \wedge t_\alpha.ST \neq t_\beta.ST)$$

Business Rule:
Two employees of the same Role, the one who lives in NYC cannot earn less than the one who does not live in NYC

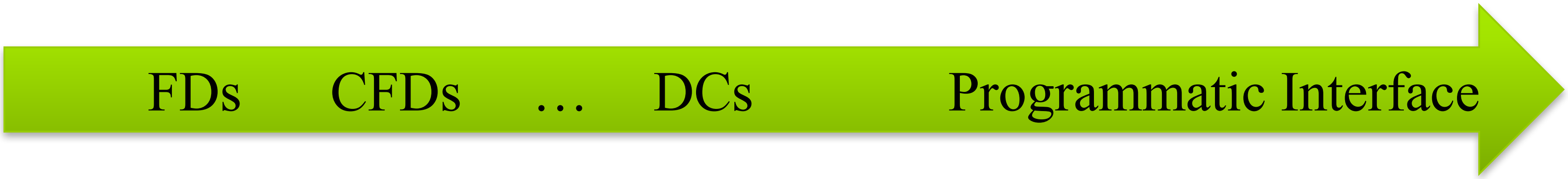
$$\begin{aligned}\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ROLE = t_\beta.ROLE \wedge t_\alpha.CITY = \text{“}NYC\text{”} \\ \wedge t_\beta.CITY \neq \text{“}NYC\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)\end{aligned}$$

Other ICs

- CINDs [Ma et al, TCS 2014]
- Metric Functional Dependencies [Koudas et al, ICDE 2009]
- Dependable Fixes
  - Editing Rules [Fan et al, VLDB 2010]
  - Fixing Rules [Wang and Tang, SIGMOD 2014]
  - Sherlock Rules [Interlandi and Tang, ICDE 2015]

Constraint Languages

Language expressiveness



FDs CFDs … DCs Programmatic Interface

Reasoning and discovery complexity

Integrity Constraints Discovery

- Schema Driven
  - Usually sensitive to the size of the schema
  - Good for long thin tables!
- Instance Driven
  - Usually sensitive to the size of the data
  - Good for fat short tables!
- Hybrid
  - Try to get the best of both worlds

Integrity Constraints Discovery

- FD Discovery:
  - TANE: Schema-driven
    - [Huhtala et al, Computer Journal 1999]
  - FASTFD: Instance-driven
    - [Wyss et al, DaWaK, 2001]
  - Hybrid
    - [Papenbrock et al, SIGMOD 2016]
- DC Discovery:
  - FASTDC: Instance-driven [Chu et al, VLDB 2013]

Integrity Constraints Discovery

- FD Discovery:
  - TANE: Schema-driven
    - [Huhtala et al, Computer Journal 1999]
  - FASTFD: Instance-driven
    - [Wyss et al, DaWaK, 2001]
  - Hybrid
    - [Papenbrock et al, SIGMOD 2016]

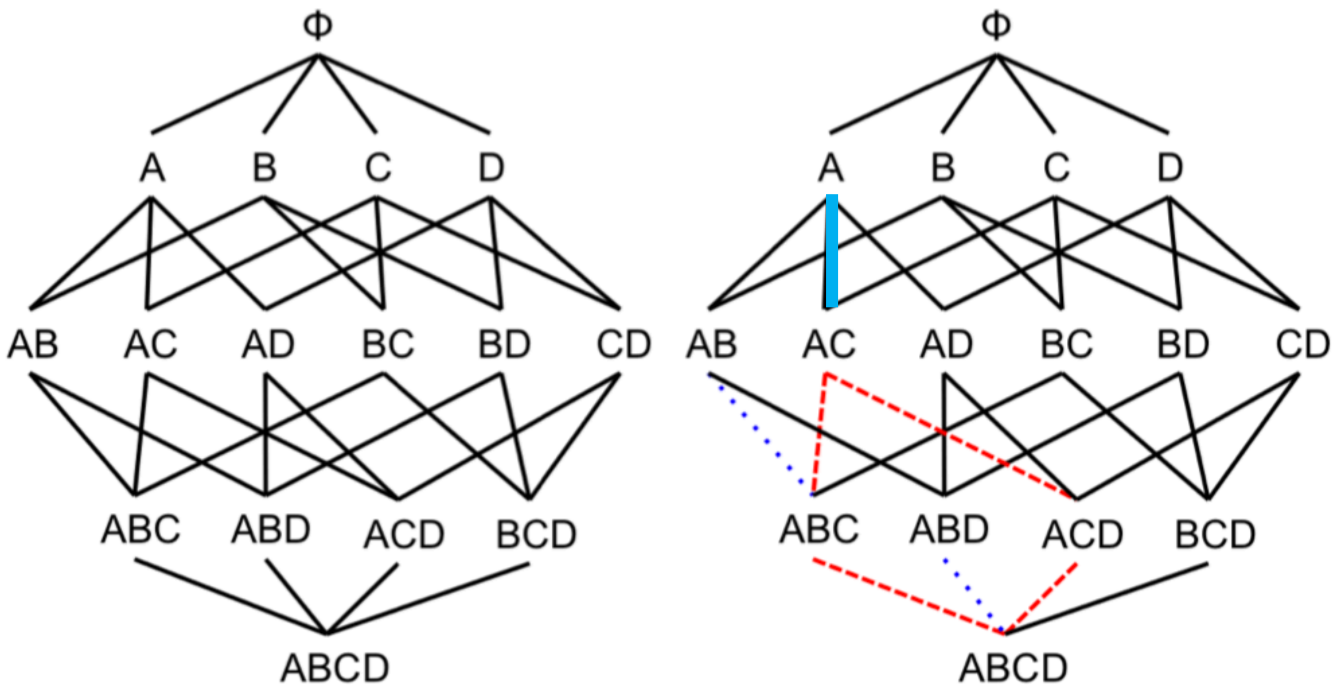
- DC Discovery:
  - FASTDC: Instance-driven [Chu et al, VLDB 2013]

FD Discovery

- Given a relational instance I of schema R, where |R| = m, find (all) **minimal, non-trivial** FDs that are **valid** on I. An FD is
  - Valid on I if there does not exist two tuples that violate the FD
  - Minimal if removing an attribute from its LHS makes it invalid
  - Trivial if the RHS is a subset of the LHS
- We want FDs with only one attribute in RHS

TANE [Huhtala et al, Computer Journal 1999]

- Generate space of FDs



(a) Space of FDs.

(b) Candidate FDs pruned if $A \rightarrow C$ is valid

TANE

- FD Validation

$$\Pi_X = \{\{t_1\}, \{t_2, t_3\}, \{t_4\}\}$$

$$\Pi_Y = \{\{t_1, t_2, t_3\}, \{t_4\}\}$$

$$\Pi_{XY} = \{\{t_1\}, \{t_2, t_3\}, \{t_4\}\}$$

$X \rightarrow Y$ is a valid FD if and only if

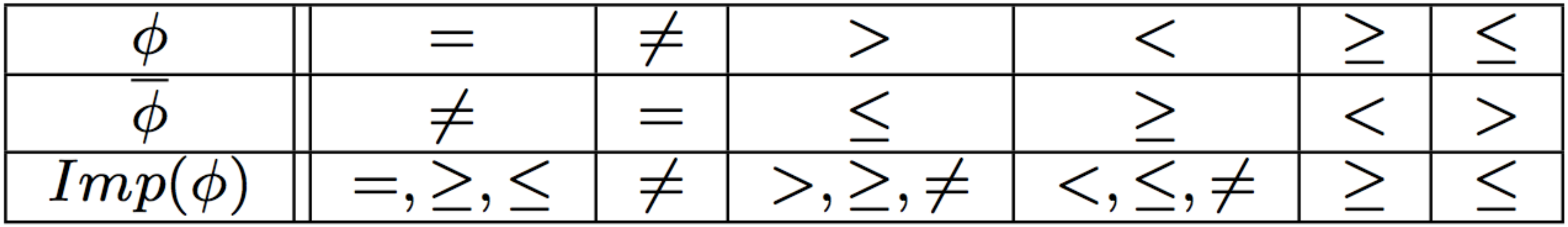
$$|\Pi_X| = |\Pi_{X \cup Y}|$$

DC Discovery: Axioms

Triviality

$\forall P_i, P_j,\ if\ P_i \in Imp(P_j)\ then \neg(\bar{P}_i \wedge P_j)$ is a trivial DC

$$\forall t_\alpha, t_\beta \in R, \neg(\boxed{t_\alpha.SAL = t_\beta.SAL} \wedge \boxed{t_\alpha.SAL > t_\beta.SAL})$$



ϕ	$=$	$\neq$	$>$	$<$	$\geq$	$\leq$
$\overline{\phi}$	$\neq$	$=$	$\leq$	$\geq$	$<$	$>$
$Imp(\phi)$	$=, \geq, \leq$	$\neq$	$>, \geq, \neq$	$<, \leq, \neq$	$\geq$	$\leq$

DC Discovery: Axioms

Augmentation

$$\mathit{If}\ \neg(P_1 \wedge \ldots \wedge P_n)\ \mathit{is\ valid},\ \mathit{then}\ \neg(P_1 \wedge \ldots \wedge P_n \wedge Q)\ \mathit{is\ also\ valid}$$

Not Minimal

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$$

$$\Downarrow$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST \wedge \boxed{t_\alpha.SAL < t_\beta.SAL)}$$

DC Discovery: Axioms

Transitivity (more like composition)

If $\neg(P_1 \wedge \ldots \wedge P_n \wedge Q_1)$, *and* $\neg(R_1 \wedge \ldots \wedge R_m \wedge Q_2)$ *are valid, and* $Q_2 \in Imp(\overline{Q_1})$, *then* $\neg(P_1 \wedge \ldots \wedge P_n \wedge R_1 \wedge \ldots \wedge R_m)$ *is valid*

$$\forall t_\alpha, t_\beta \in R, \neg(\overset{Q_1}{\boxed{t_\alpha.ST = t_\beta.ST}} \wedge t_\alpha.SAL < t_\beta.SAL \wedge t_\alpha.TR > t_\beta.TR)$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge \overset{Q_2}{\boxed{t_\alpha.ST \neq t_\beta.ST}})$$

$$\Downarrow$$

$$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.SAL < t_\beta.SAL \wedge t_\alpha.TR > t_\beta.TR)$$

DC Discovery

Given a relational schema R and an instance I, find all **non-trivial**, **minimal** DCs that **hold on I**

Focus on DCs involving at most two tuples

FASTDC [Chu et al, VLDB 2013]

Employee

TID	I(String)	M(String)	S(Double)
t_1	A1	A1	50
t_2	A2	A1	40
t_3	A3	A1	40

- The space of predicates

$P_1 : t_\alpha.I = t_\beta.I$

$P_2 : t_\alpha.I \neq t_\beta.I$

$P_3 : t_\alpha.M = t_\beta.M$

$P_4 : t_\alpha.M \neq t_\beta.M$

$P_5 : t_\alpha.S = t_\beta.S$

$P_6 : t_\alpha.S \neq t_\beta.S$

$P_7 : t_\alpha.S > t_\beta.S$

$P_8 : t_\alpha.S \leq t_\beta.S$

$P_9 : t_\alpha.S < t_\beta.S$

$P_{10} : t_\alpha.S \geq t_\beta.S$

$P_{11} : t_\alpha.I = t_\alpha.M$

$P_{12} : t_\alpha.I \neq t_\alpha.M$

$P_{13} : t_\alpha.I = t_\beta.M$

$P_{14} : t_\alpha.I \neq t_\beta.M$

- Any combination of predicates constitutes a candidate DC

FASTDC

$\neg(P_i \wedge P_j \wedge P_k)$ is a valid DC w.r.t. I

⇕

For every tuple pair in I , P_i, P_j, P_k cannot be true together

⇕

For every tuple pair in I , at least one of P_i, P_j, P_k is false

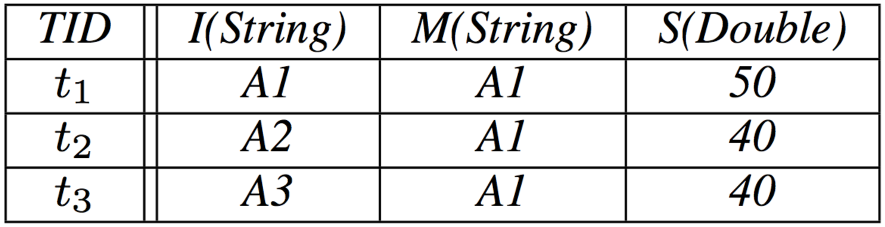
⇕

For every tuple pair in I , at least one of $\overline{P_i}, \overline{P_j}, \overline{P_k}$ is true

⇕

$\overline{P_i}, \overline{P_j}, \overline{P_k}$ covers the ***set of true predicates for every tuple pair***

FASTDC



TID	I(String)	M(String)	S(Double)
t_1	A1	A1	50
t_2	A2	A1	40
t_3	A3	A1	40

Evi$_I$

$< t_2, t_3 >, < t_3, t_2 > \{P_2, P_3, P_5, P_8, P_{10}, P_{12}, P_{14}\}$

$< t_2, t_1 >, < t_3, t_1 > \{P_2, P_3, P_6, P_8, P_9, P_{12}, P_{14}\}$

$< t_1, t_2 >, < t_1, t_3 > \{P_2, P_3, P_6, P_7, P_{10}, P_{11}, P_{13}\}$

$\{P_{10}, P_{14}\}$ covers the set of true predicates for every tuple pair

$$\forall t_\alpha, t_\beta \in R, \neg(\overline{P}_{10} \wedge \overline{P}_{14})$$

$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.S < t_\beta.S \wedge t_\alpha.I = t_\beta.M)$ is a valid DC

$\{P_5, P_{10}, P_{14}\}$ covers the set of true predicates for every tuple pair

$\neg(\overline{P_{10}} \wedge \overline{P_{14}} \wedge \overline{P_5})$ is a valid DC, but not minimal

FASTDC

Evi$_I$ $\{P_2, P_3, P_5, P_8, P_{10}, P_{12}, P_{14}\}$

$\{P_2, P_3, P_6, P_8, P_9, P_{12}, P_{14}\}$

$\{P_2, P_3, P_6, P_7, P_{10}, P_{11}, P_{13}\}$

$P_8 \in Imp(\overline{P}_6)$

$P_{10} \in Imp(\overline{P}_6)$

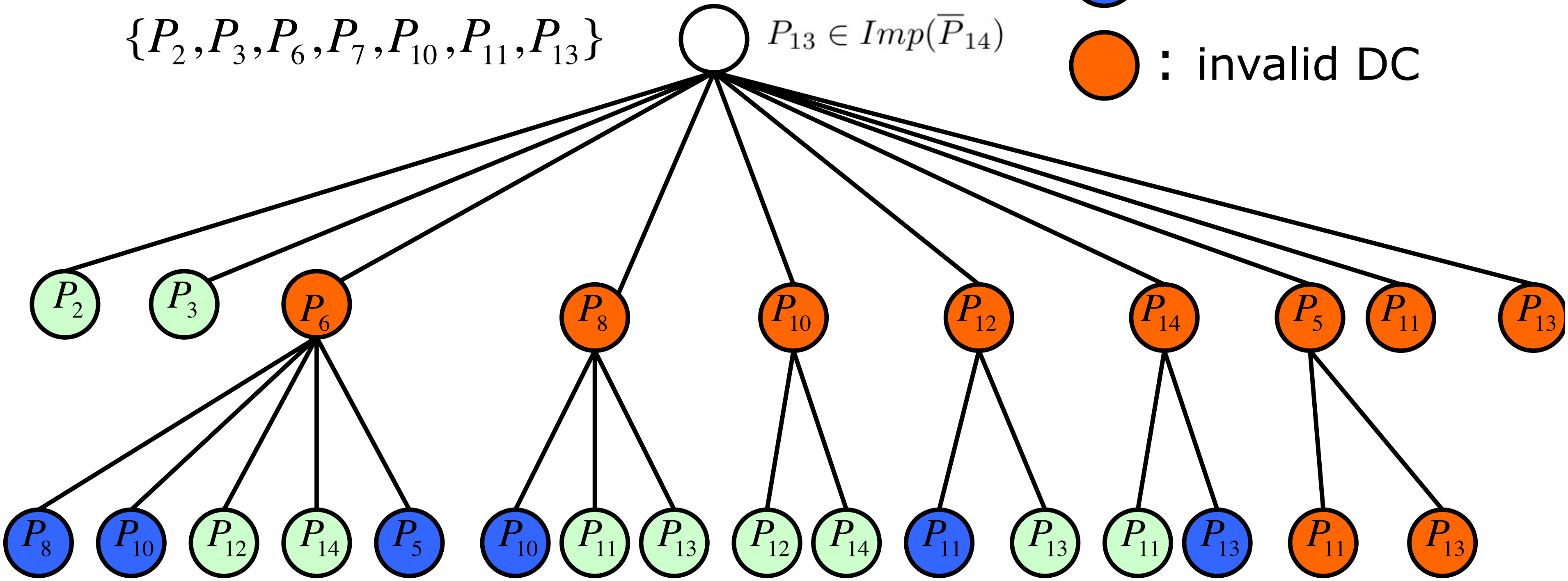
$P_{11} \in Imp(\overline{P}_{12})$

$P_{13} \in Imp(\overline{P}_{14})$

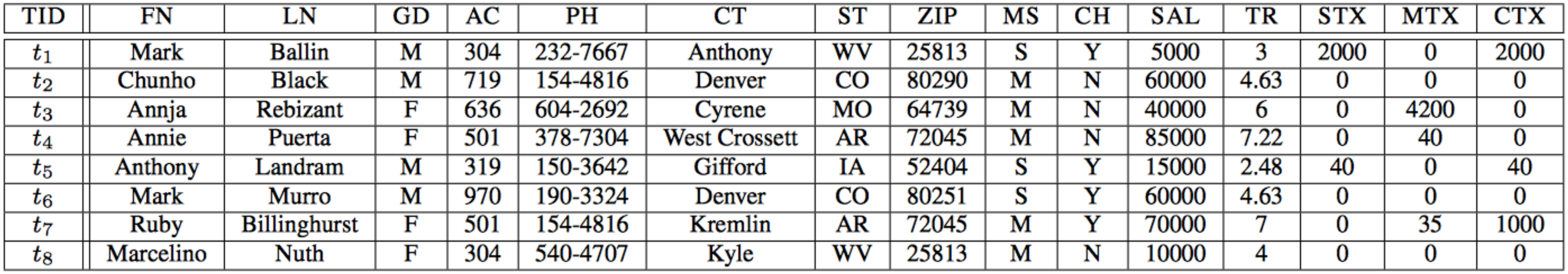
🟢 : valid DC

🔵 : pruned branch

🟠 : invalid DC



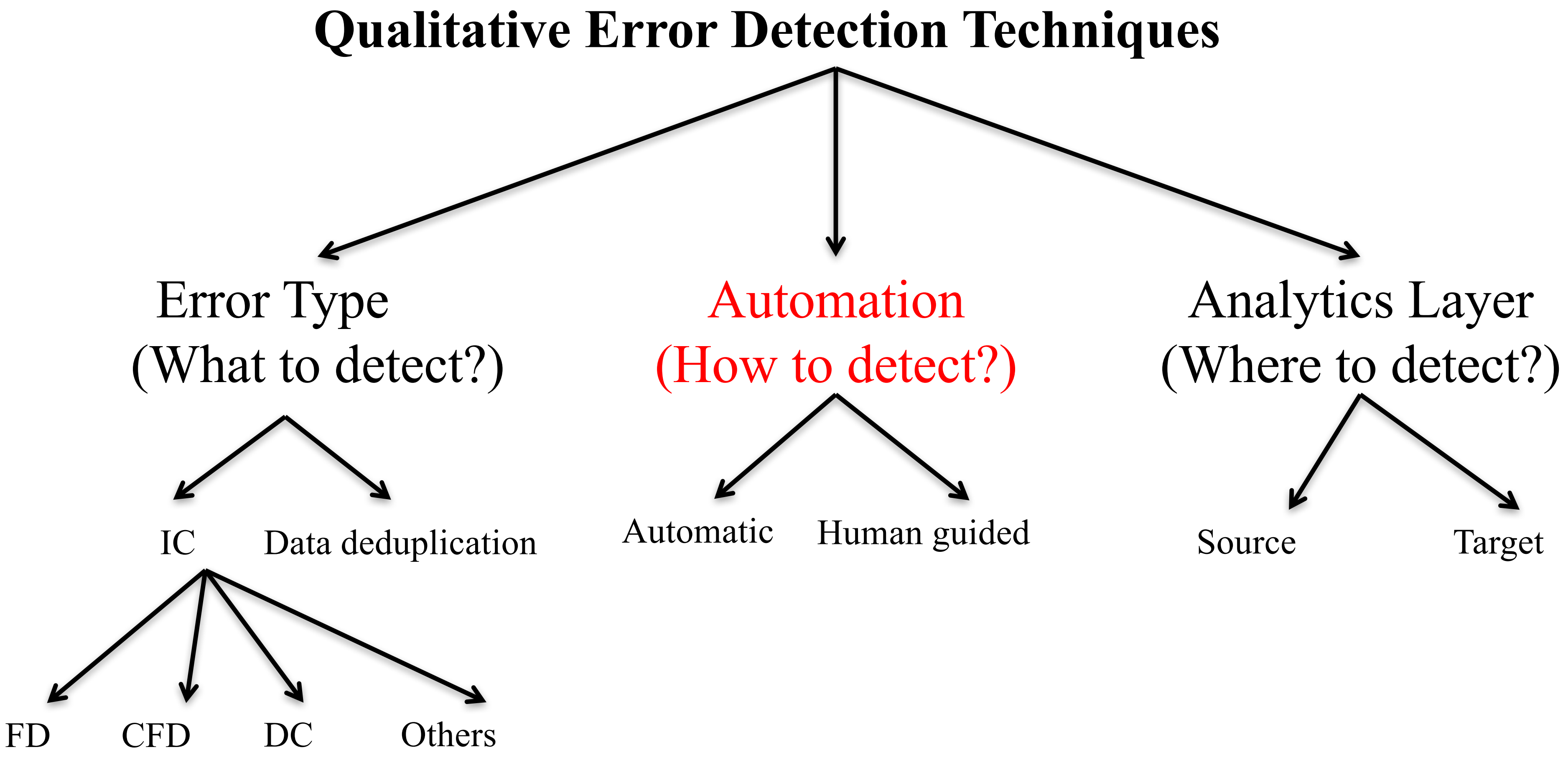
FASTDC



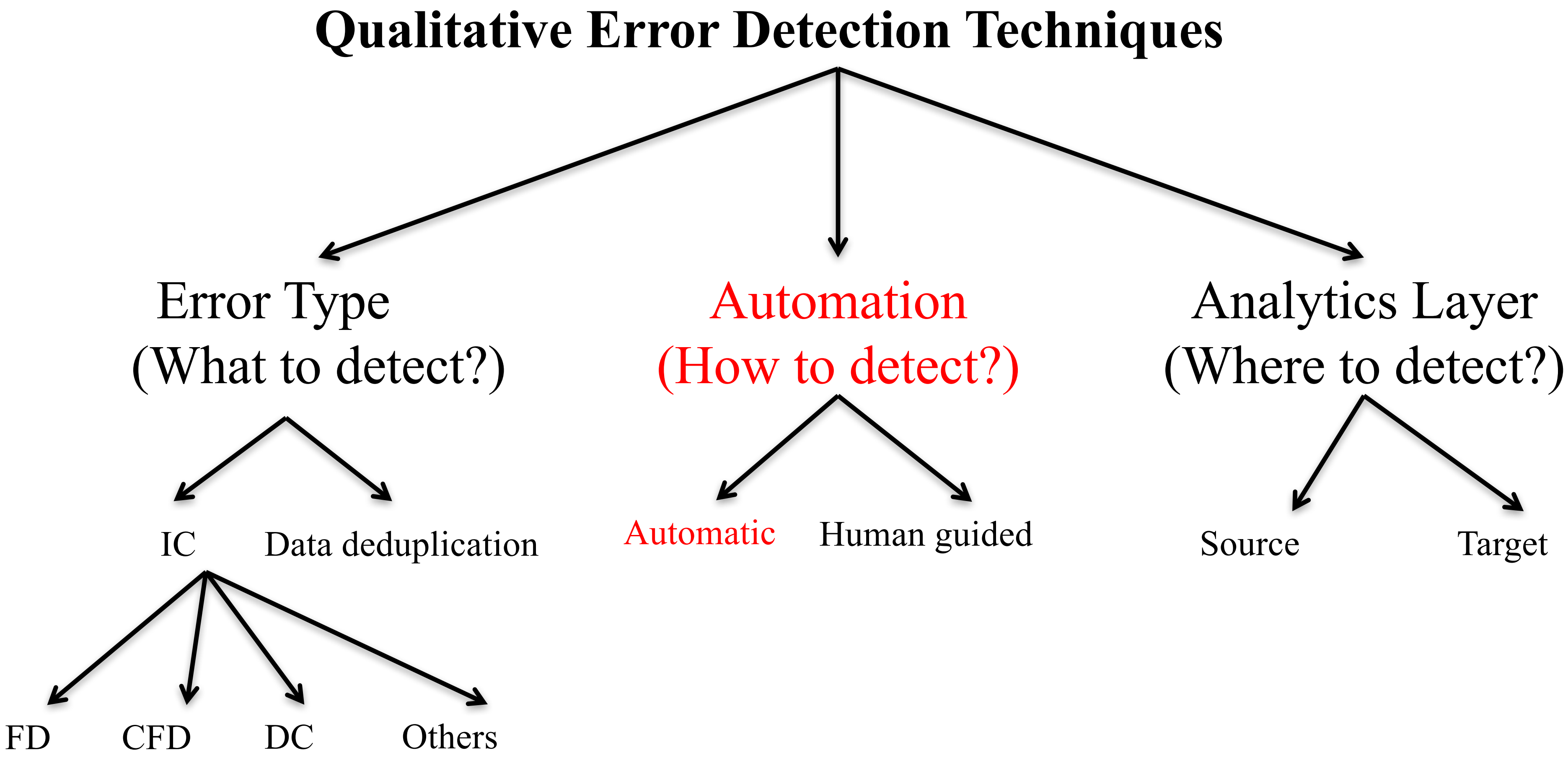
TID	FN	LN	GD	AC	PH	CT	ST	ZIP	MS	CH	SAL	TR	STX	MTX	CTX
t_1	Mark	Ballin	M	304	232-7667	Anthony	WV	25813	S	Y	5000	3	2000	0	2000
t_2	Chunho	Black	M	719	154-4816	Denver	CO	80290	M	N	60000	4.63	0	0	0
t_3	Annja	Rebizant	F	636	604-2692	Cyrene	MO	64739	M	N	40000	6	0	4200	0
t_4	Annie	Puerta	F	501	378-7304	West Crossett	AR	72045	M	N	85000	7.22	0	40	0
t_5	Anthony	Landram	M	319	150-3642	Gifford	IA	52404	S	Y	15000	2.48	40	0	40
t_6	Mark	Murro	M	970	190-3324	Denver	CO	80251	S	Y	60000	4.63	0	0	0
t_7	Ruby	Billinghurst	F	501	154-4816	Kremlin	AR	72045	M	Y	70000	7	0	35	1000
t_8	Marcelino	Nuth	F	304	540-4707	Kyle	WV	25813	M	N	10000	4	0	0	0

$Key:\{AC,PH\}$	$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.AC = t_\beta.AC \wedge t_\alpha.PH = t_\beta.PH)$
$Domain: MS \in \{S,M\}$	$\forall t_\alpha \in R, \neg(t_\alpha.MS \neq S \wedge t_\alpha.MS \neq M)$
$FD: ZIP \rightarrow ST$	$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ZIP = t_\beta.ZIP \wedge t_\alpha.ST \neq t_\beta.ST)$
$CFD: CT = Los\ Angeles \rightarrow ST = CA$	$\forall t_\alpha \in R, \neg(t_\alpha.CT = Los\ \text{Angeles} \wedge \text{t}_\alpha.\text{ST} \neq CA)$
$Check: SAL \geq STX$	$\forall t_\alpha \in R, \neg(t_\alpha.SAL < t_\alpha.STX)$
Business logic	$\forall t_\alpha, t_\beta \in R, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.SAL < t_\beta.SAL \wedge t_\alpha.TR > t_\beta.TR)$

Error Detection Techniques Taxonomy



Error Detection Techniques Taxonomy



Qualitative Error Detection Techniques

Error Type
(What to detect?)

Automation
(How to detect?)

Analytics Layer
(Where to detect?)

IC

Data deduplication

Automatic

Human guided

Source

Target

FD

CFD

DC

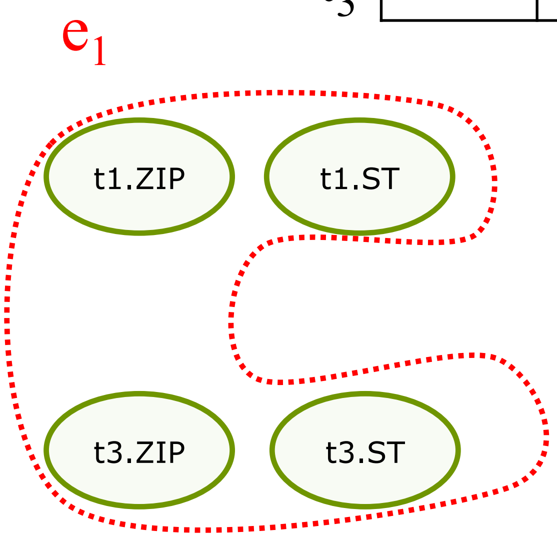
Others

Holistic Error Detection

- Vertex: Cell in the database
- Hyperedge: A set of cells that violate a DC

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85376	NY	110
t_2	211	Mark	White	M	90012	NY	80
t_3	386	Mark	Lee	E	85376	AZ	75

Employee Table



Zip → ST

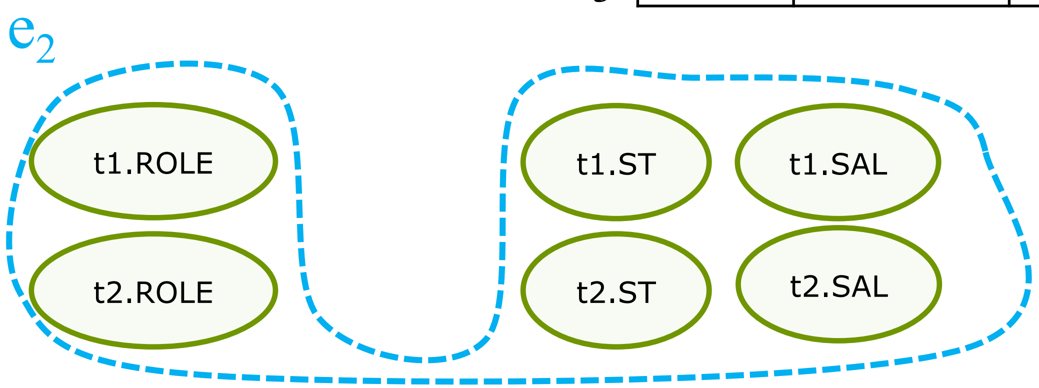
[Chu et al, ICDE 2013]
[Kolahi and Lakshmanan ICDT 2009]

Holistic Error Detection

- Vertex: Cell in the database
- Hyperedge: A set of cells that violate a DC

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85376	NY	110
t_2	211	Mark	White	M	90012	NY	80
t_3	386	Mark	Lee	E	85376	AZ	75

Employee Table



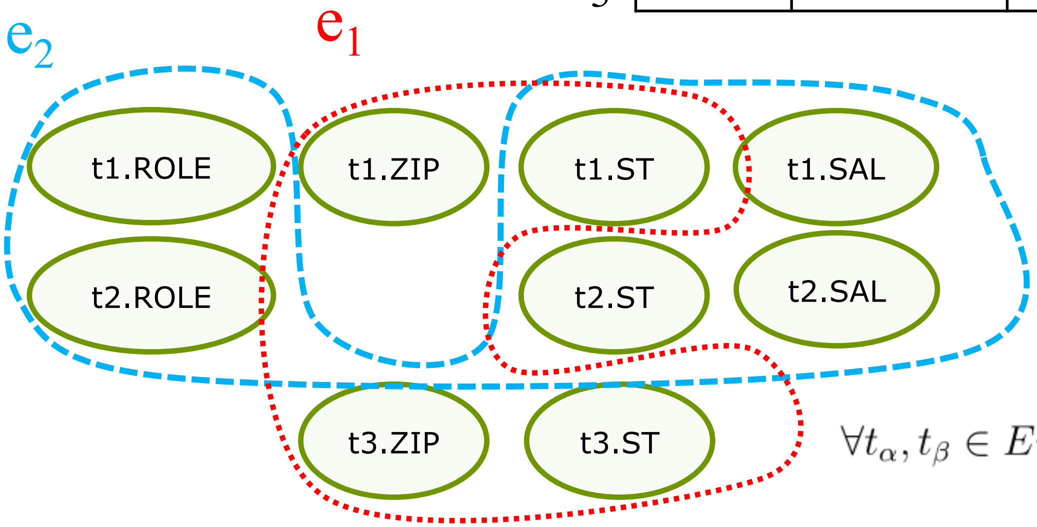
$$\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“}M\text{”} \wedge t_\beta.ROLE = \text{“}E\text{”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Holistic Error Detection

- Vertex: Cell in the database
- Hyperedge: A set of cells that violate a DC

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85376	NY	110
t_2	211	Mark	White	M	90012	NY	80
t_3	386	Mark	Lee	E	85376	AZ	75

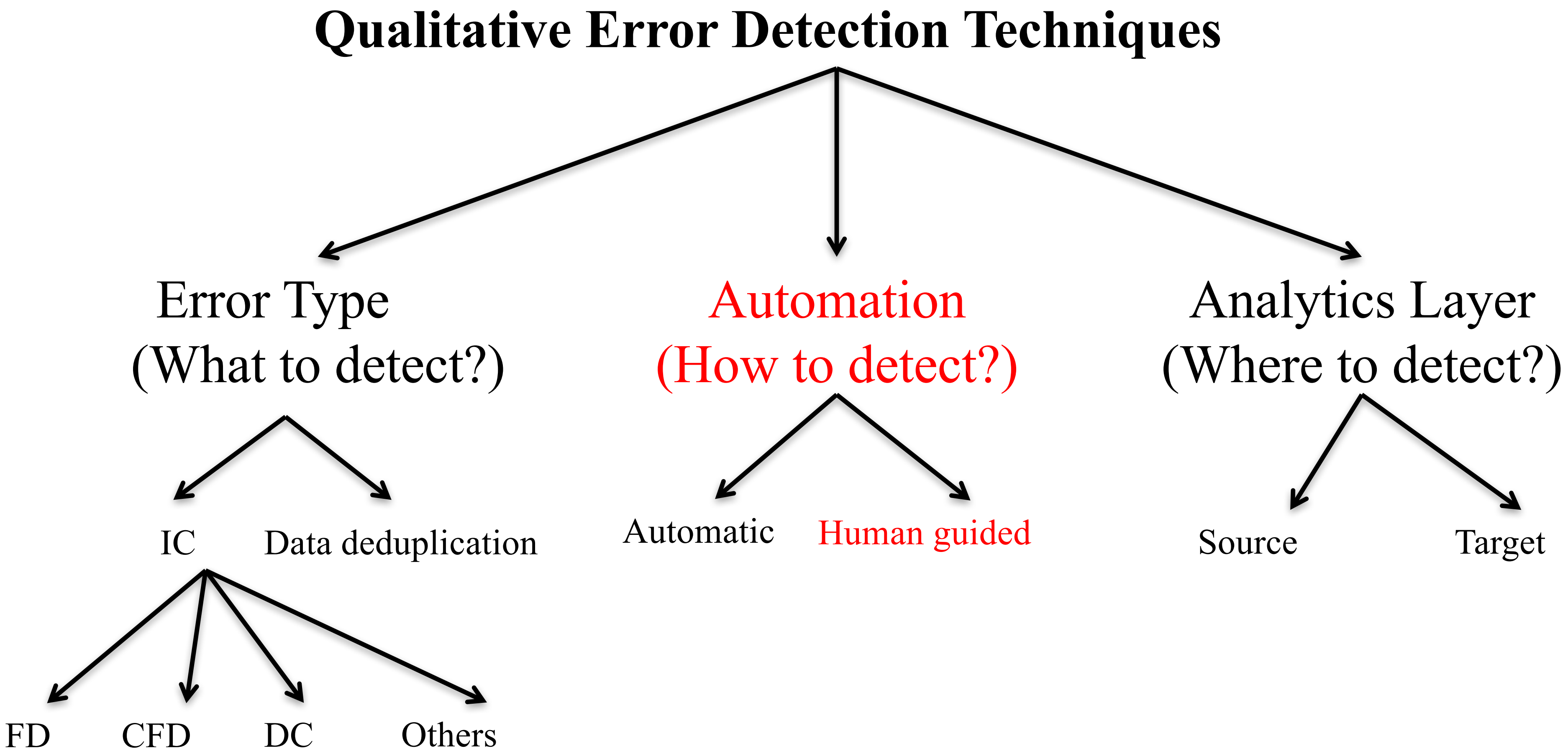
Employee Table



Zip → ST

$$\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = “M” \wedge t_\beta.ROLE = “E” \wedge t_\alpha.SAL < t_\beta.SAL)$$

Error Detection Techniques Taxonomy



Qualitative Error Detection Techniques

Error Type
(What to detect?)

Automation
(How to detect?)

Analytics Layer
(Where to detect?)

IC

Data deduplication

Automatic

Human guided

Source

Target

FD

CFD

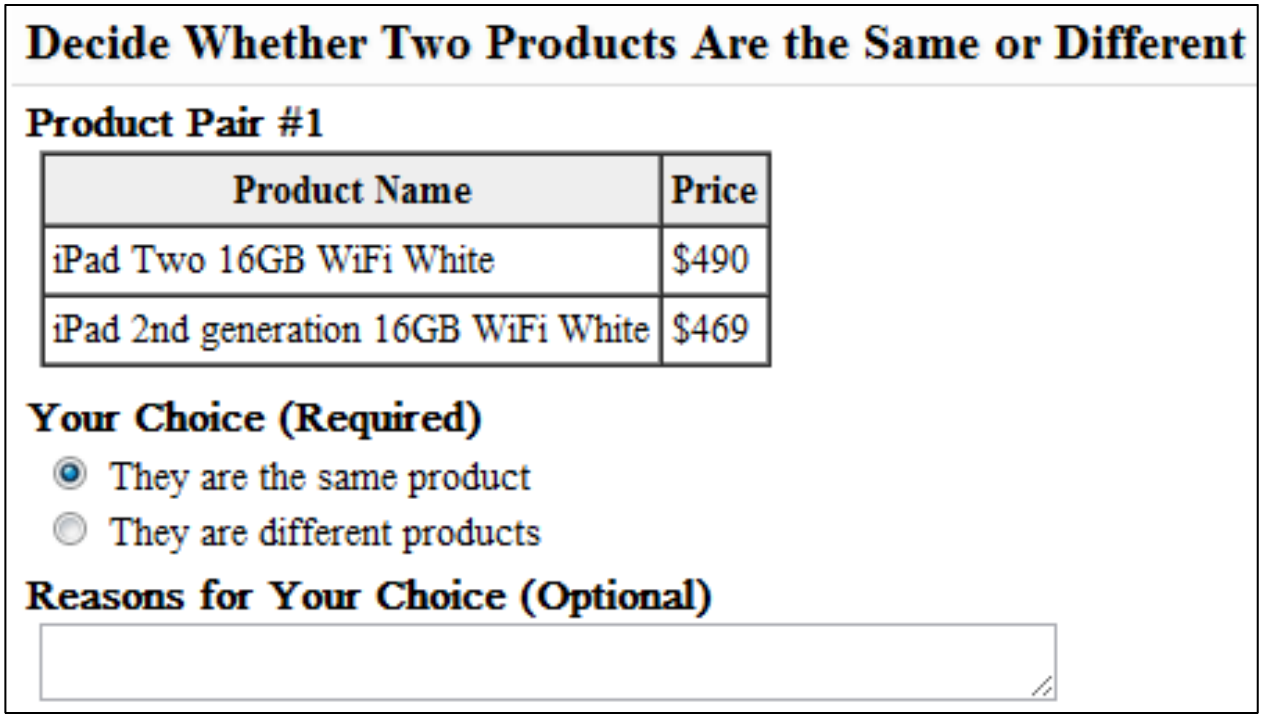
DC

Others

CrowdER: [Wang et al, VLDB 2012]

- Human-Intelligence Task (HIT)

$O(n^2)$ X



Decide Whether Two Products Are the Same or Different

Product Pair #1

Product Name	Price
iPad Two 16GB WiFi White	$490
iPad 2nd generation 16GB WiFi White	$469

Your Choice (Required)

● They are the same product

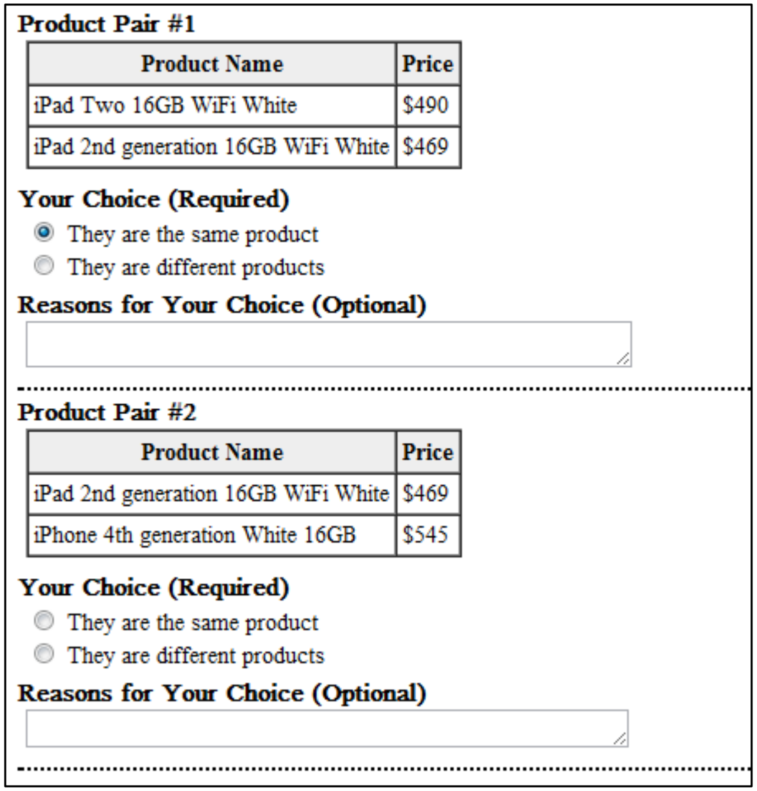
○ They are different products

Reasons for Your Choice (Optional)

CrowdER: Batching Strategies

- Pair-based HIT

$O(n^2/k)$ X



Product Pair #1

Product Name	Price
iPad Two 16GB WiFi White	$490
iPad 2nd generation 16GB WiFi White	$469

Your Choice (Required)

● They are the same product

○ They are different products

Reasons for Your Choice (Optional)

Product Pair #2

Product Name	Price
iPad 2nd generation 16GB WiFi White	$469
iPhone 4th generation White 16GB	$545

Your Choice (Required)

○ They are the same product

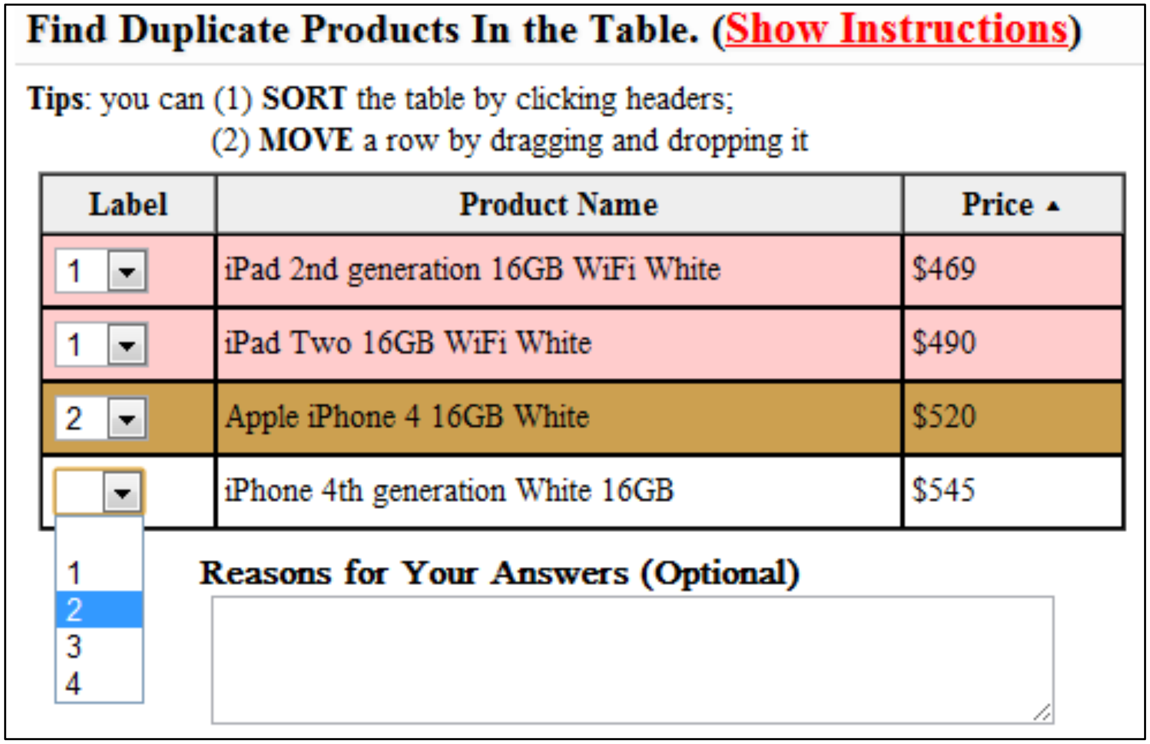
○ They are different products

Reasons for Your Choice (Optional)

CrowdER: Batching Strategies

- Cluster-based HIT

$O(n^2/k^2)$ X



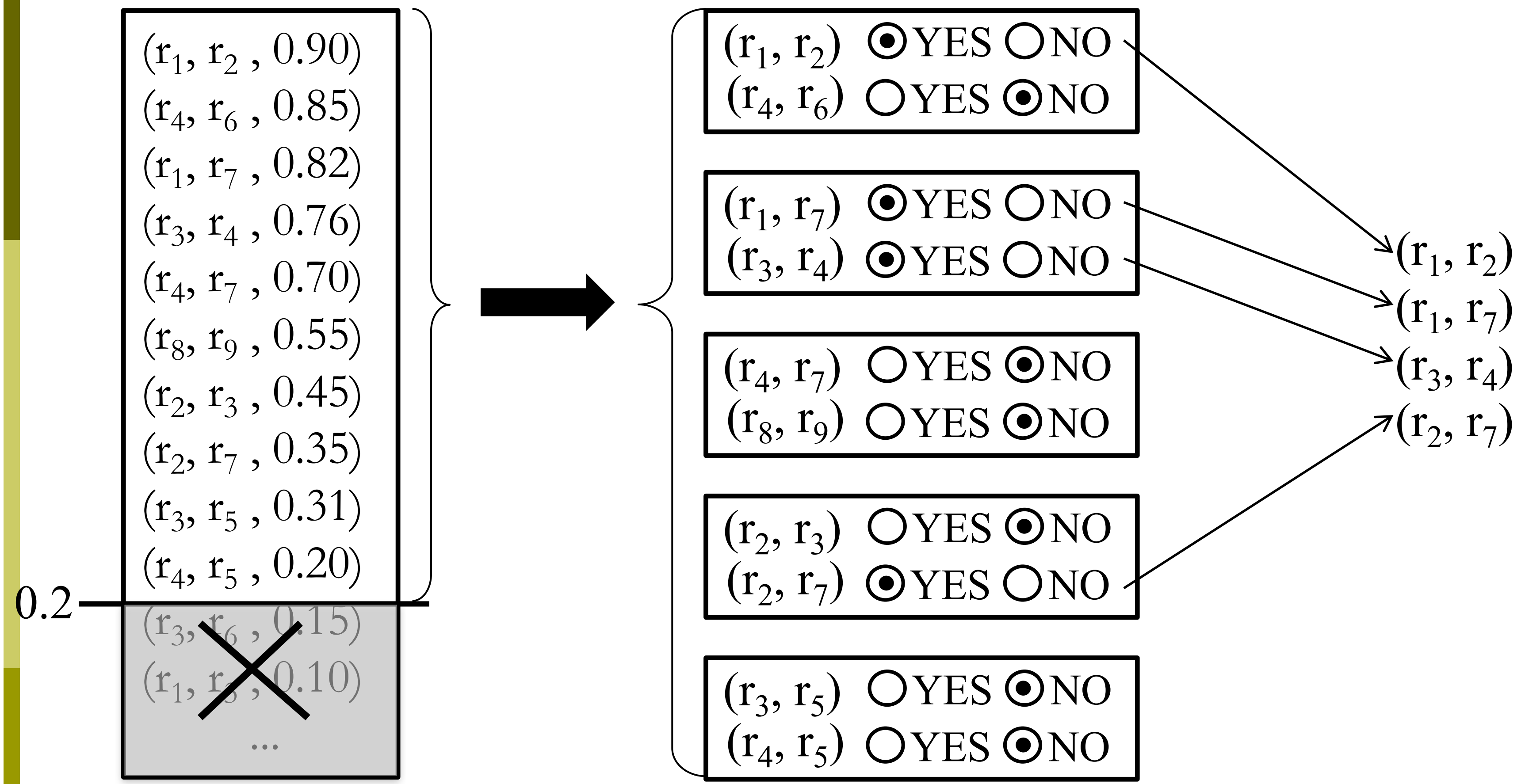
Find Duplicate Products In the Table. (Show Instructions)

Tips: you can (1) **SORT** the table by clicking headers;
(2) **MOVE** a row by dragging and dropping it

Label	Product Name	Price ▲
1	iPad 2nd generation 16GB WiFi White	$469
1	iPad Two 16GB WiFi White	$490
2	Apple iPhone 4 16GB White	$520
	iPhone 4th generation White 16GB	$545

Reasons for Your Answers (Optional)

CrowdER: Workflow



$(r_1, r_2, 0.90)$
$(r_4, r_6, 0.85)$
$(r_1, r_7, 0.82)$
$(r_3, r_4, 0.76)$
$(r_4, r_7, 0.70)$
$(r_8, r_9, 0.55)$
$(r_2, r_3, 0.45)$
$(r_2, r_7, 0.35)$
$(r_3, r_5, 0.31)$
$(r_4, r_5, 0.20)$
0.2
$(r_3, r_6, 0.15)$
$(r_1, r_3, 0.10)$
…

(r_1, r_2) ● YES ○ NO
(r_4, r_6) ○ YES ● NO

(r_1, r_7) ● YES ○ NO
(r_3, r_4) ● YES ○ NO

(r_4, r_7) ○ YES ● NO
(r_8, r_9) ○ YES ● NO

(r_2, r_3) ○ YES ● NO
(r_2, r_7) ● YES ○ NO

(r_3, r_5) ○ YES ● NO
(r_4, r_5) ○ YES ● NO

(r_1, r_2)
(r_1, r_7)
(r_3, r_4)
(r_2, r_7)

(a) Remove the pairs whose likelihoods < 0.2

(b) Generate HITs to verify the pairs of records

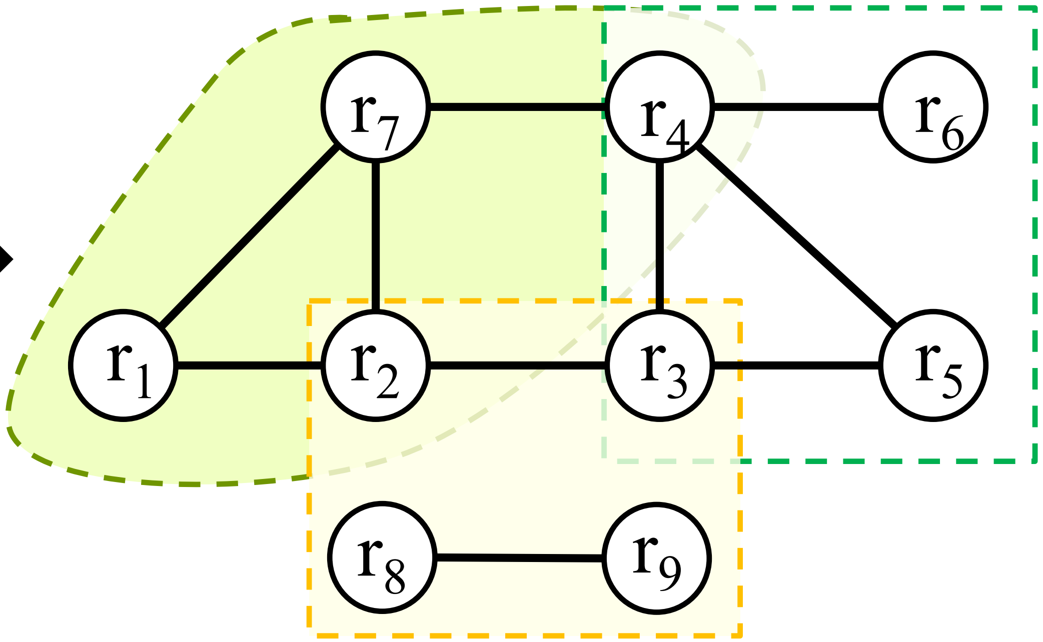
(c) Output matching pairs

CrowdER: Workflow

(r_1, r_2 , 0.90)
(r_4, r_6 , 0.85)
(r_1, r_7 , 0.82)
(r_3, r_4 , 0.76)
(r_4, r_7 , 0.70)
(r_8, r_9 , 0.55)
(r_2, r_3 , 0.45)
(r_2, r_7 , 0.35)
(r_3, r_5 , 0.31)
(r_4, r_5 , 0.20)

0.2

~~(r_3, r_6 , 0.15)~~
~~(r_1, r_3 , 0.10)~~
...



Cluster-based HIT

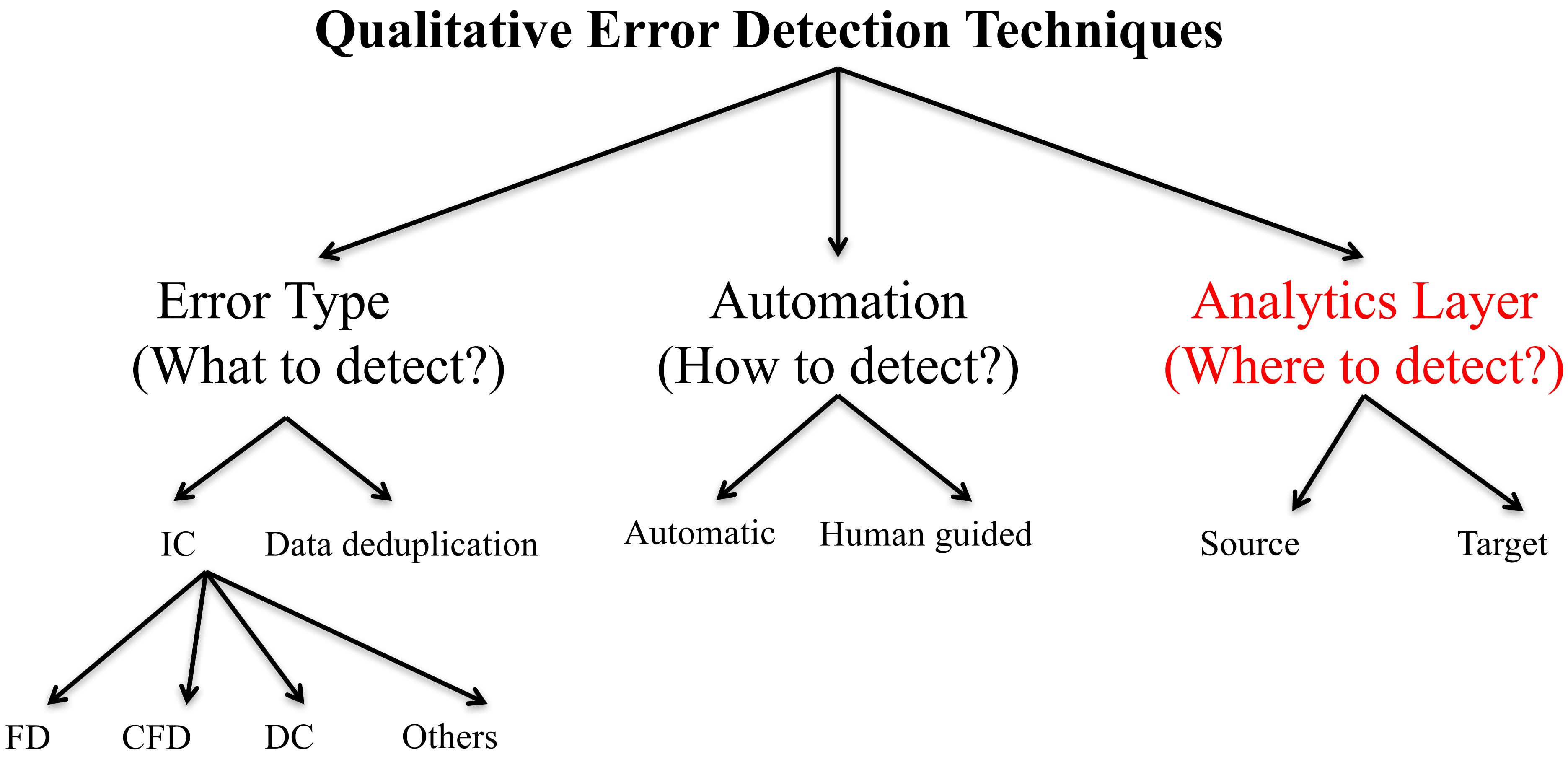
Cluster-size threshold k

Minimize the number of HITs

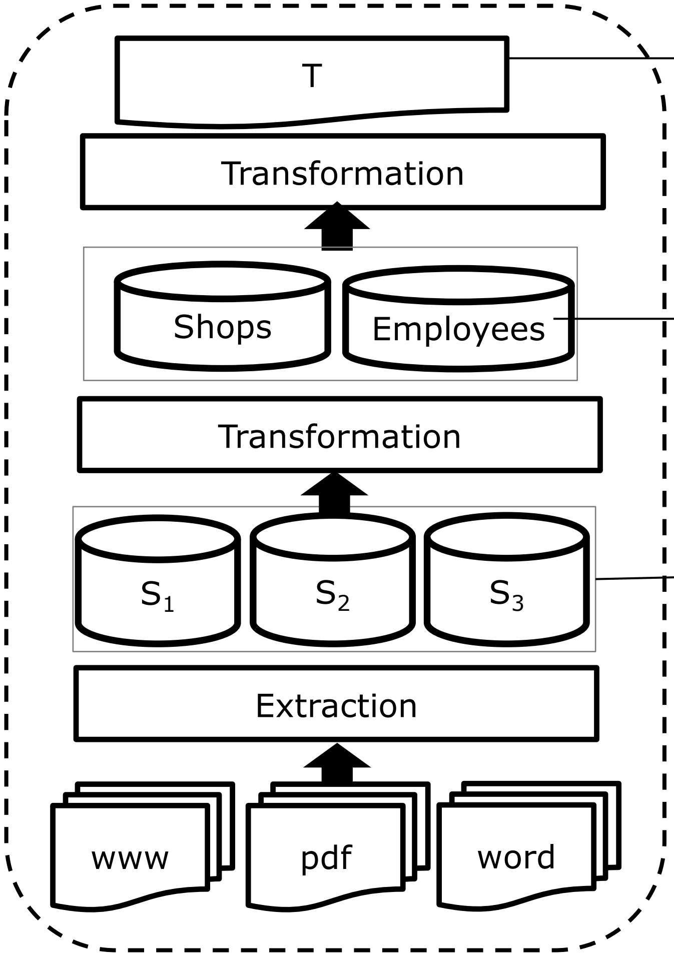
NP-Hard

HIT_1	HIT_2	HIT_3
r_1	r_3	r_2
r_2	r_4	r_3
r_4	r_5	r_8
r_7	r_6	r_9

Error Detection Techniques Taxonomy



Decoupled in Space and Time



(1) In the same shop, the average salary for the managers (Grd=2) should be higher than the one for the staff (Grd=1)
(2) A bigger shop cannot have a smaller number of staff

(3) Phone number must have country code and local number

(4) S1.NAME is NOT NULL
(5) length(S3.NAME) < 30

$$\neg(t_\alpha.Shop = t_\beta.Shop \wedge t_\alpha.Avgsal > t_\beta.Avgsal \wedge t_\alpha.Grd < t_\beta.Grd)$$

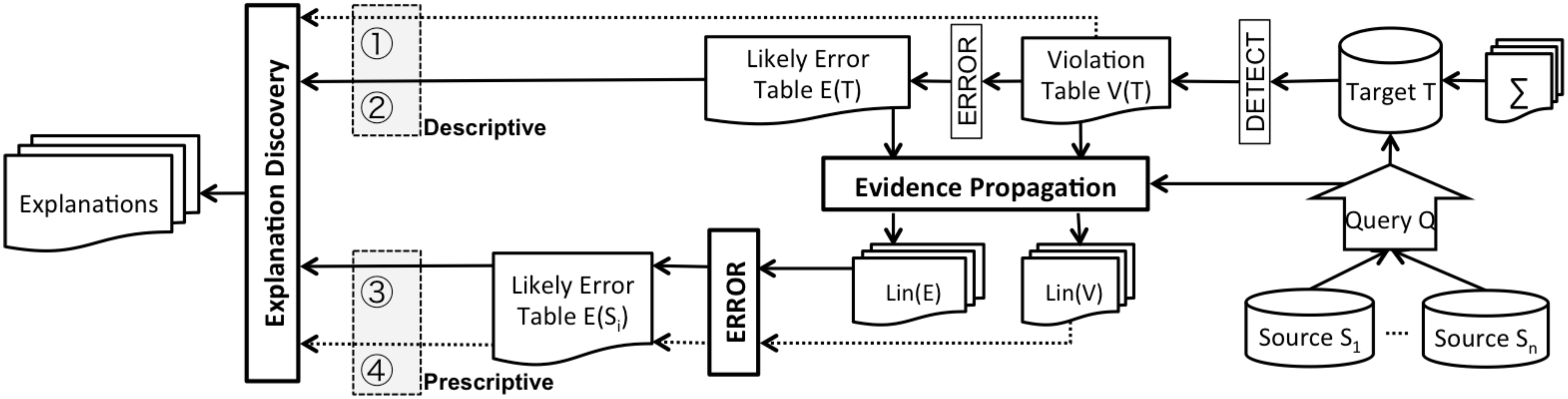
$$\neg(t_\alpha.Size > t_\beta.Size \wedge t_\alpha.\#Emps < t_\beta.\#Emps)$$

Calls for a New Solution

		Errors	
		Target	Source
Constraints	Target	Traditional Data Repair Algorithms	**Descriptive and Prescriptive Data Cleaning**
	Source	Dependency Propagation	Traditional Data Repair Algorithms

- DBRx: [Chalamalla et al., SIGMOD 2014]
- DataXRay: [Wang et al., SIGMOD 2015]
- QOCO: [Bergman et al., VLDB 2015]

DBRx Architecture [Chalamalla et al, SIGMOD 2014]



University of Waterloo

Technical Challenges

- **Errors Propagation**
  - Blowup (e.g., Aggregates)
  - Propagation Level (violations vs Fixes)
  - Distributing Responsibilities
- **Source Error Identification**
  - Assign Weights based on Query and Error Semantics
  - Accumulate Evidences (different Violation Semantics)
- **Explain Errors**

Tracing the Sources of Errors

T	Shop	Size	Grd	AvgSal	#Emps	Region
t_a	NY1	46 ft^2	2	99 $	1	US
t_b	NY1	46 ft^2	1	100 $	3	US
t_c	NY2	62 ft^2	2	96 $	2	US
t_d	NY2	62 ft^2	1	90 $	2	US
t_e	LA1	35 ft^2	2	105 $	2	US
t_f	LND	38 ft^2	1	65 £	2	EU

```
SELECT  Shops.SId as Shop, Size,
     Emps.Grd, AVG(Emps.Sal) as
     AvgSal, COUNT(EId) as  #Emps, 'US'
     as Region
FROM  US.Emps JOIN US.Shops ON Sid
GROUP BY  SId, Size, Grd
```

Emps	EId	Name	Dept	Sal	Grd	SId	JoinYr
t_1	e4	John	S	91	1	NY1	2012
t_2	e5	Anne	D	99	2	NY1	2012
t_3	e7	Mark	S	93	1	NY1	2012
t_4	e8	Claire	S	116	1	NY1	2012
t_5	e11	Ian	R	89	1	NY2	2012
t_6	e13	Laure	R	94	2	NY2	2012
t_7	e14	Mary	E	91	1	NY2	2012
t_8	e18	Bill	D	98	2	NY2	2012
t_9	e14	Mike	R	94	2	LA1	2011
t_{10}	e18	Claire	E	116	2	LA1	2011

Shops	SId	City	State	Size	Start
t_{11}	NY1	NYC	NY	*46 ft^2*	2011
t_{12}	NY2	NYC	NY	*62 ft^2*	2012
t_{13}	LA1	LA	CA	35 ft^2	2011

2?

Average salary of higher grade in the same shop should be higher!

Error Contribution Scores

Emps	EId [CSV]	Sal [CSV]	Grd [CSV]	SId[CSV]		[RSV]
t_1	e4 [‘’,$\frac{1}{3}$]	91 [$\frac{91}{300}$,‘’]	1 [$\frac{1}{3}$, $\frac{1}{3}$]	NY1 [$\frac{1}{3}$,‘’]		[0,1]
t_2	e5	99 [0,‘’]	2 [1,‘’]	NY1 [1,‘’]		[1,‘’]
t_3	e7 [‘’,$\frac{1}{3}$]	93 [$\frac{93}{300}$,‘’]	1 [$\frac{1}{3}$, $\frac{1}{3}$]	NY1 [$\frac{1}{3}$,‘’]		[0,1]
t_4	e8 [‘’,$\frac{1}{3}$]	116 [$\frac{116}{300}$,‘’]	1 [$\frac{1}{3}$, $\frac{1}{3}$]	NY1 [$\frac{1}{3}$,‘’]		[1,1]
t_5	e11 [‘’,$\frac{1}{2}$]	89	1 [‘’,$\frac{1}{2}$]	NY2		[‘’,0]
t_6	e13	94	2	NY2		[]
t_7	e14 [‘’,$\frac{1}{2}$]	91	1 [‘’,$\frac{1}{2}$]	NY2		[‘’,0]
t_8	e18	98	2	NY2		[]
t_9	e14	94	2	LA1		[]
t_{10}	e18	116	2	LA1		[]

$cs_v(c)$:

Contribution of this cell to the aggregate

Shops	SId [CSV]	Size [CSV]		[RSV]
t_{12}	NY1 [2,‘’]	46 [‘’,1]		[1,1]
t_{13}	NY2	62 [‘’,1]		[‘’,1]
t_{14}	LA1	35		[]

$rs_v(t)$:

Removing t_4 eliminates the violations

Identifying Likely Errors

- Maximize a gain function of adding more source errors

$$Gain(H_v) = \sum_{s \in H_v} c_v(s) - \sum_{1 \leq j \leq |H_v|} \sum_{j < k \leq |H_v|} D(s_j, s_k)$$

$$D(s_j, s_k) = |c_v(s_j) - c_v(s_k)| \qquad c_v(s) = cs_v(s) + rs_v(s)$$

tid	Score
s_1	0.67
s_2	0.54
s_3	0.47
s_4	0.08
s_5	0.06
s_6	0.05

Gain = 1.08

→

tid	Score
s_1	0.67
s_2	0.54
s_3	0.47
s_4	0.08
s_5	0.06
s_6	0.05

Gain = 1.28

→

tid	Score
s_1	0.67
s_2	0.54
s_3	0.47
s_4	0.08
s_5	0.06
s_6	0.05

Gain = -0.08

Error Explanation

Likely Error Tuples

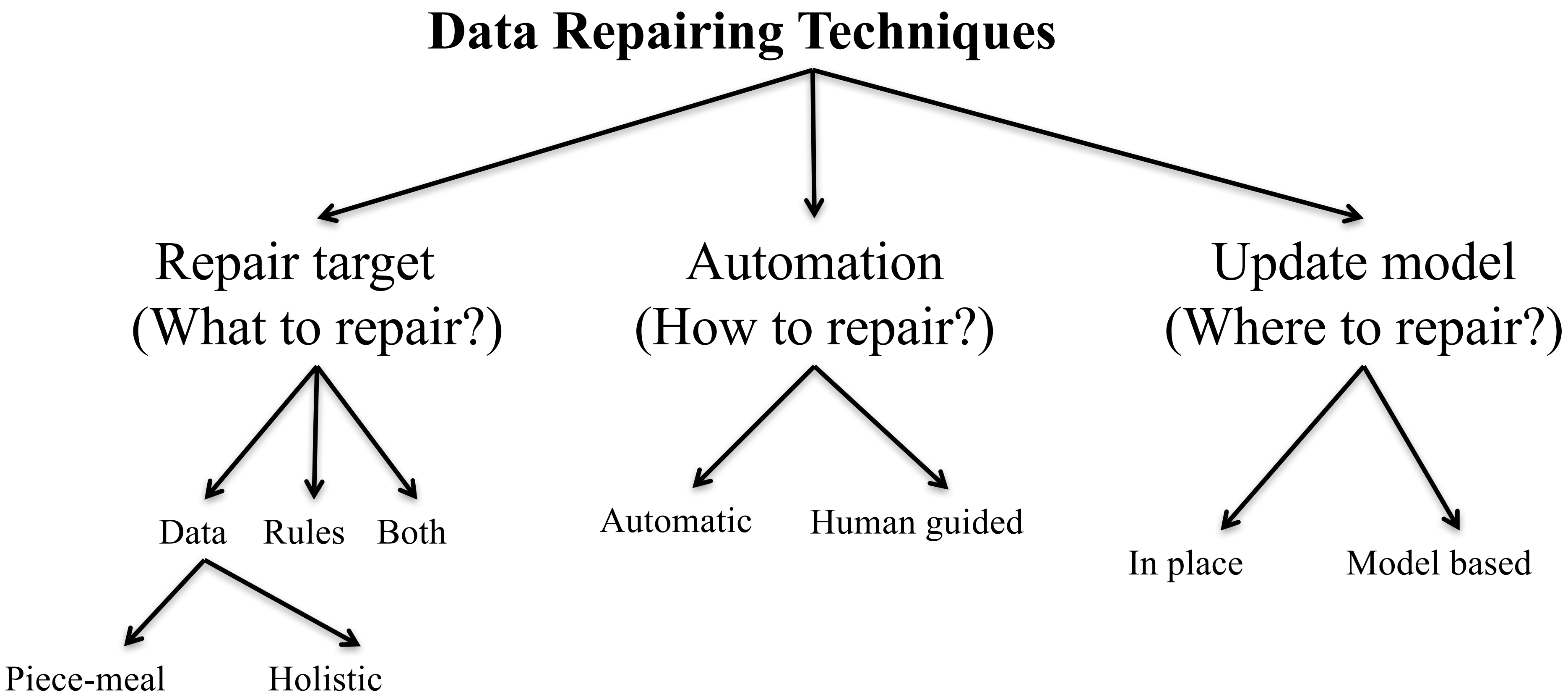
Emps	EId	Name	Dept	Sal	Grd	SId	JoinYr
t_1	e4	John	S	91	1	NY1	2012
t_2	e5	Anne	D	99	2	NY1	2012
t_3	e7	Mark	S	93	1	NY1	2012
t_4	e8	Claire	S	116	1	NY1	2012
t_5	e11	Ian	R	89	1	NY2	2012
t_6	e13	Laure	R	94	2	NY2	2012
t_7	e14	Mary	E	91	1	NY2	2012
t_8	e18	Bill	D	98	2	NY2	2012
t_9	e14	Mike	R	94	2	LA1	2011
t_{10}	e18	Claire	E	116	2	LA1	2011

Possible Explanations

Explanation	Recall	Precision	Concise
$Dept = s$	Low	High	Concise
$eid = e_4 \vee eid = e_7 \vee$ $eid = e_8 \vee eid = e_{14}$	High	High	Verbose
$Grd = 1$	High	Low	Concise

Data Repairing

Data Repairing Techniques Taxonomy



Data Repairing Techniques

Repair target (What to repair?)

Automation (How to repair?)

Update model (Where to repair?)

Data

Rules

Both

Automatic

Human guided

In place

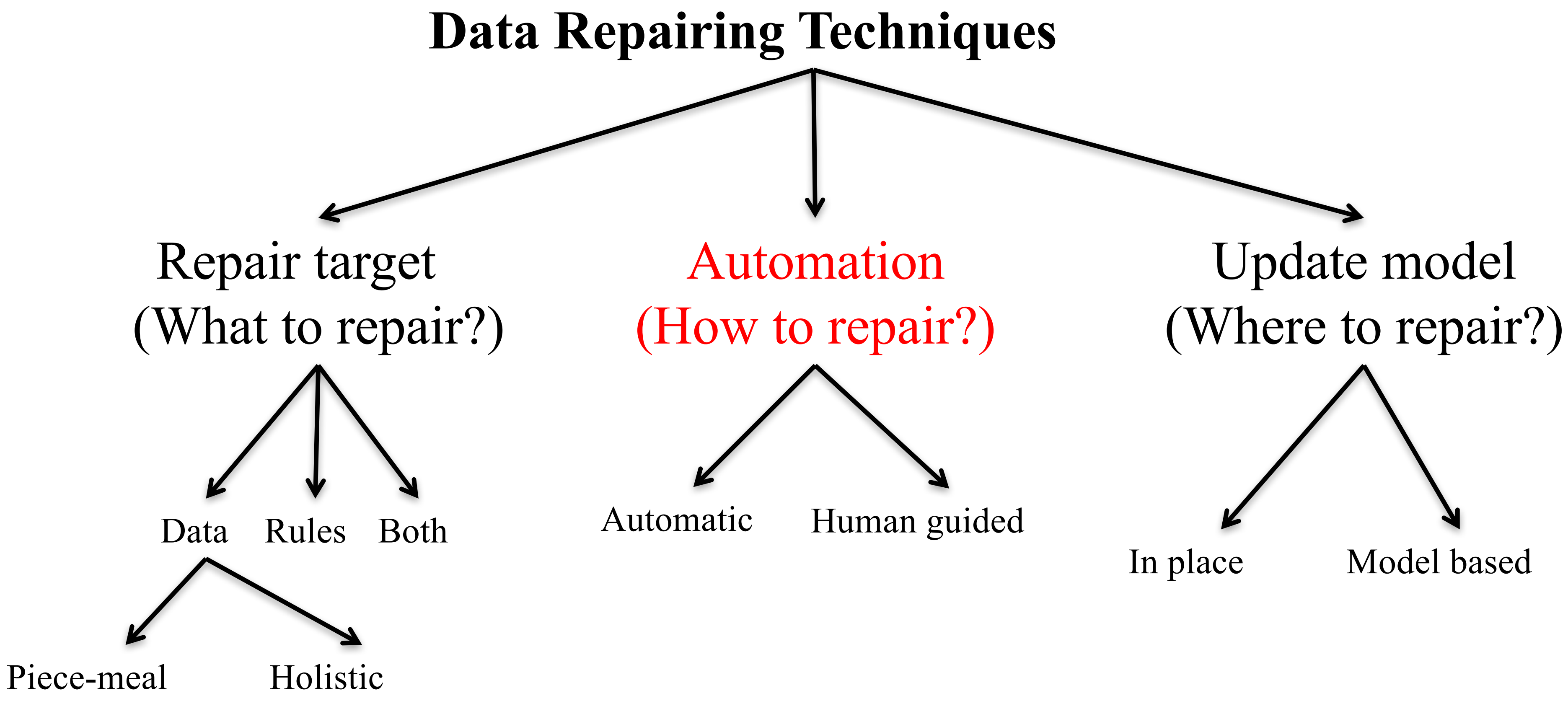
Model based

Piece-meal

Holistic

[Ilyas and Chu, Foundations and Trends in Database Systems, 2015]

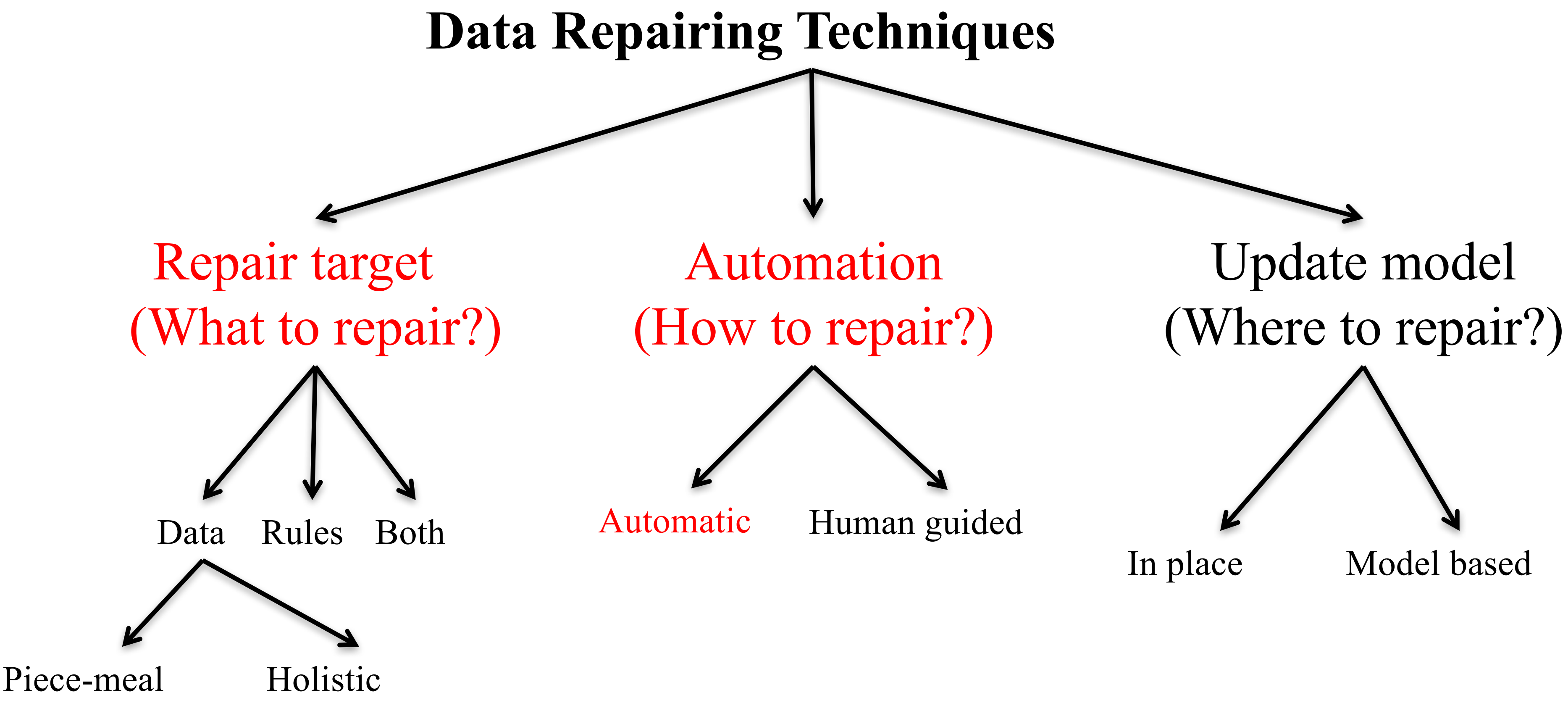
Data Repairing Techniques Taxonomy



Repair Automation

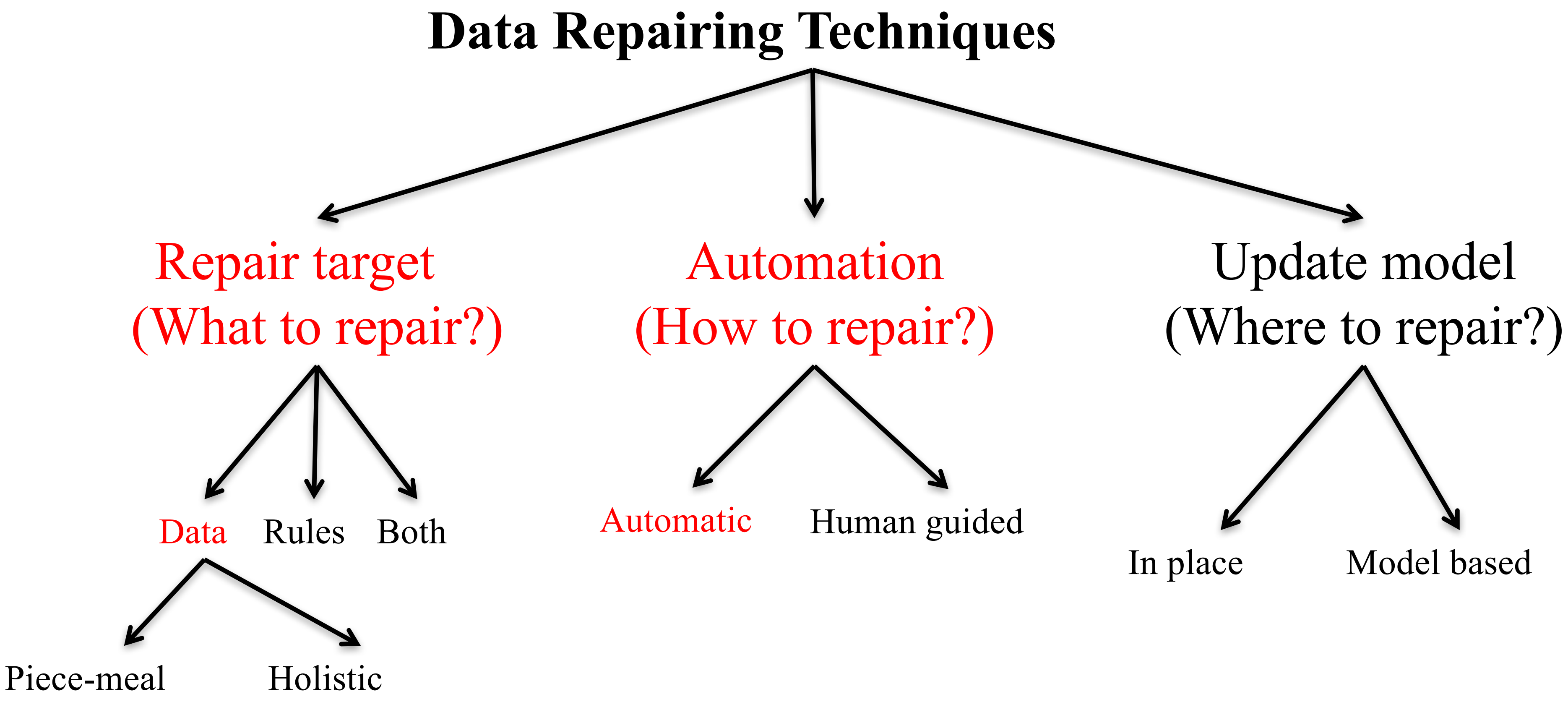
- Most automatic repairing techniques adopt the “minimality” of repairs principle
- Repairing techniques in practice are predominantly manual and semi-automatic at best
- Will survey both

Data Repairing Techniques Taxonomy



University of Waterloo

Data Repairing Techniques Taxonomy



Data Repairing Techniques

Repair target (What to repair?)

Automation (How to repair?)

Update model (Where to repair?)

Data

Rules

Both

Automatic

Human guided

In place

Model based

Piece-meal

Holistic

Data Repair by Value Update

- I is a dirty database if $I \nvDash \Sigma$, and I_j is a repair for I if $I_j \vDash \Sigma$
- For a repair I_j, $\Delta(I_j)$ is the set of changed cells in I_j

I

	A	B
t_1	1	2
t_2	1	3
t_3	1	3
t_4	4	5

$\Sigma = \{A \rightarrow B\}$

I_1

	A	B
t_1	1	3
t_2	1	3
t_3	1	3
t_4	4	5

$\Delta(I_1) = \{t_1[B]\}$

I_2

	A	B
t_1	1	2
t_2	1	2
t_3	1	2
t_4	4	5

$\Delta(I_2) = \{t_2[B], t_3[B]\}$

Data Only Repairing

Cardinality-Minimal repairs

- Commonly used in obtaining a single repair automatically
- Repairs with the minimum number of changes
- I_1 is Card-Min iff $\nexists I_2$ s.t. $|\Delta(I_2)| < |\Delta(I_1)|$

	A	B
t_1	1	2
t_2	1	3
t_3	1	3
t_4	4	5

FD: {A→ B}

I_1

A	B
1	3
1	3
1	3
4	5

✓

I_2

A	B
1	2
1	2
1	2
4	5

✗

I_3

A	B
1	5
1	5
1	5
4	5

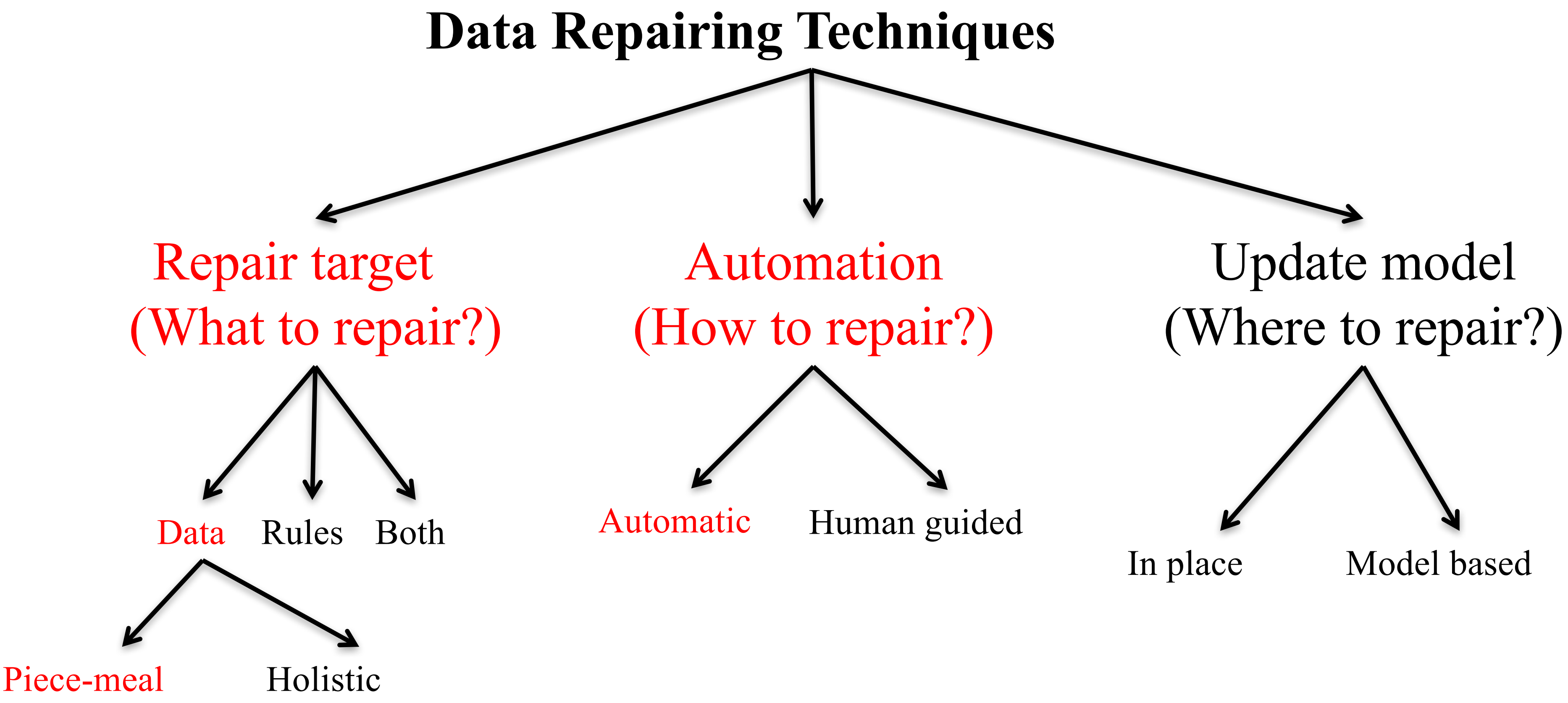
✗

I_4

A	B
7	3
1	3
1	3
4	5

✗

Data Repairing Techniques Taxonomy



Data Repairing Techniques

Repair target (What to repair?)

Automation (How to repair?)

Update model (Where to repair?)

Data

Rules

Both

Automatic

Human guided

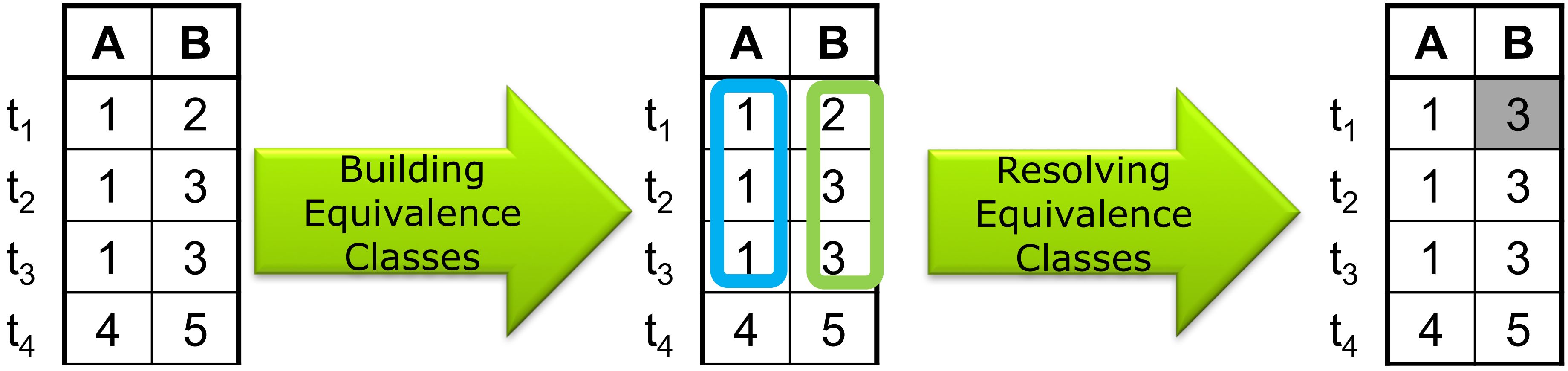
In place

Model based

Piece-meal

Holistic

FD Repairing [Bohannon et al, SIGMOD 2005]



	A	B
t_1	1	2
t_2	1	3
t_3	1	3
t_4	4	5

FD: {A→ B}

Building Equivalence Classes

	A	B
t_1	1	2
t_2	1	3
t_3	1	3
t_4	4	5

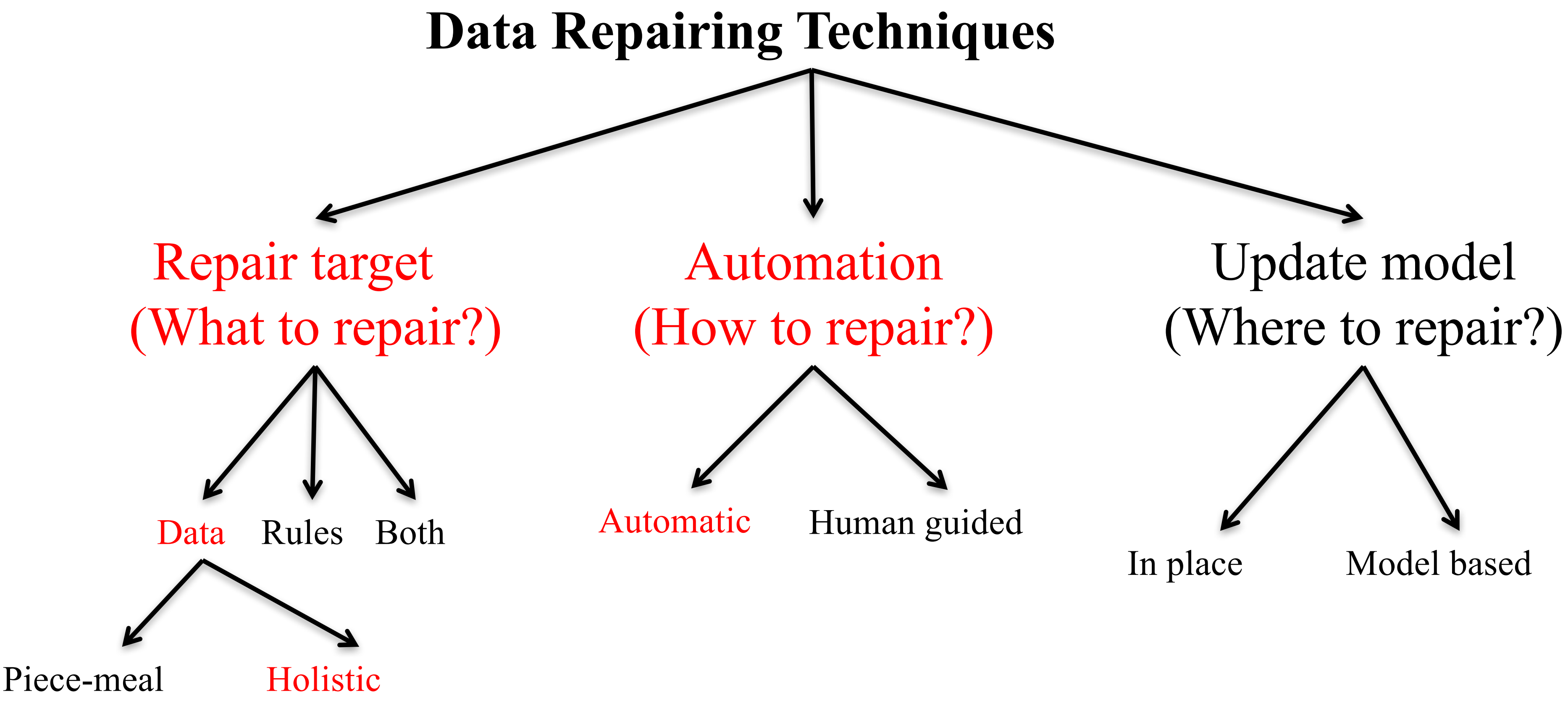
FD: {A→ B}

Resolving Equivalence Classes

	A	B
t_1	1	3
t_2	1	3
t_3	1	3
t_4	4	5

FD: {A→ B}

Data Repairing Techniques Taxonomy

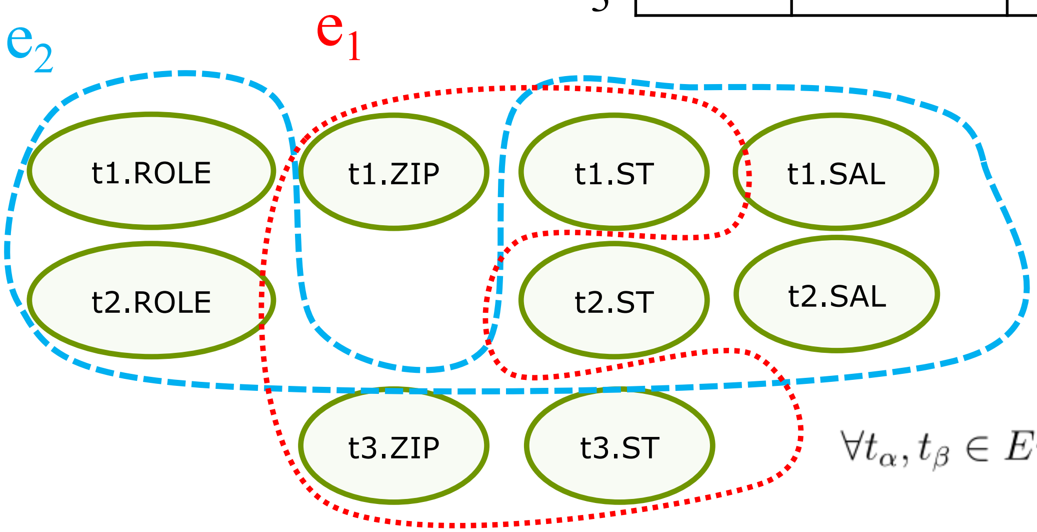


Holistic Data Repairing [Chu et al, ICDE 2013]

- Vertex: Cell in the database
- Hyperedge: A set of cells that violate a DC

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85376	NY	110
t_2	211	Mark	White	M	90012	NY	80
t_3	386	Mark	Lee	E	85376	AZ	75

Employee Table



Zip → ST

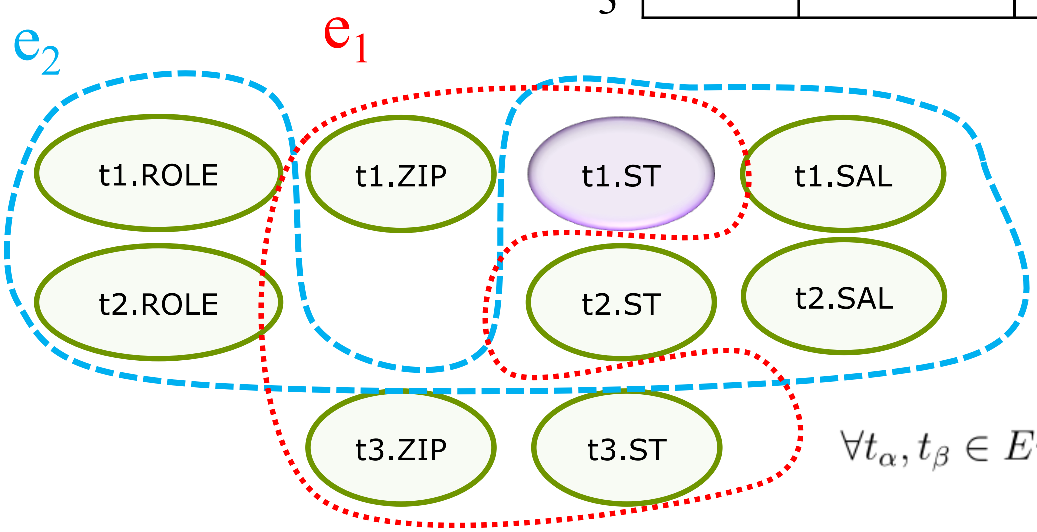
$$\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“M”} \wedge t_\beta.ROLE = \text{“E”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Step1: Minimal Vertex Cover

- A minimal set of vertices that are intersecting with every hyperedge

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85376	NY	110
t_2	211	Mark	White	M	90012	NY	80
t_3	386	Mark	Lee	E	85376	AZ	75

Employee Table

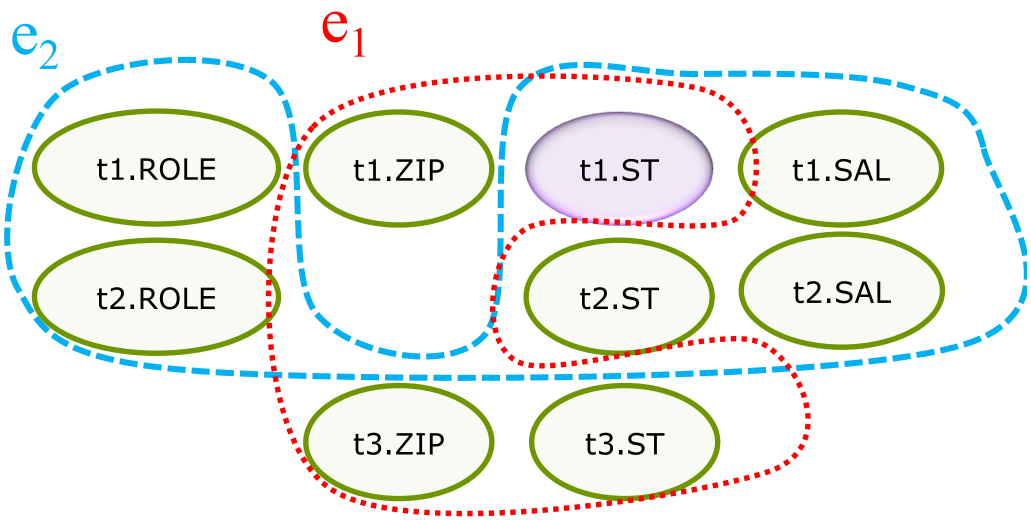


Zip → ST

$$\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = “M” \\ \wedge t_\beta.ROLE = “E” \wedge t_\alpha.SAL < t_\beta.SAL)$$

Step2: Collect Repair Requirements

- A set of conditions that need to be satisfied to resolve all violations



Condition to resolve e_1 by changing t1.ST:

$t1.ST = t3.ST$

Condition to resolve e_2 by changing t1.ST:

$t1.ST \mathrel{!}= t2.ST$

Zip → ST

$$\forall t_\alpha, t_\beta \in Emp, \neg(t_\alpha.ST = t_\beta.ST \wedge t_\alpha.ROLE = \text{“M”} \\ \wedge t_\beta.ROLE = \text{“E”} \wedge t_\alpha.SAL < t_\beta.SAL)$$

Step3: Get Updates

- A set of assignments satisfying the conditions, with minimal number of changed cells

t1.ST = t3.ST
t1.ST != t2.ST

	ID	FN	LN	ROLE	ZIP	ST	SAL
t_1	105	Anne	Nash	E	85376	NY	110
t_2	211	Mark	White	M	90012	NY	80
t_3	386	Mark	Lee	E	85376	AZ	75

AZ

Gradually increase the number of cells that are going to be changed, until reach a solution

Suppose we only want to change t1.ST
t2.ST = NY
t3.ST = AZ

More Holistic Data Repairing [Fan et al, SIGMOD 2011]

Tran

FN	LN	St	City	AC	Post	Phn	Item
Robert	Brady	5 Wren St	Ldn	020	WC1H 9SE	3887644	watch
Robert	Brady	5 Wren St	Ldn	020	WC1E 7HX	3887644	necklace

Master: Card

FN	LN	St	City	AC	Zip	Tel
Robert	Brady	5 Wren St	Ldn	020	WC1H 9SE	3887644

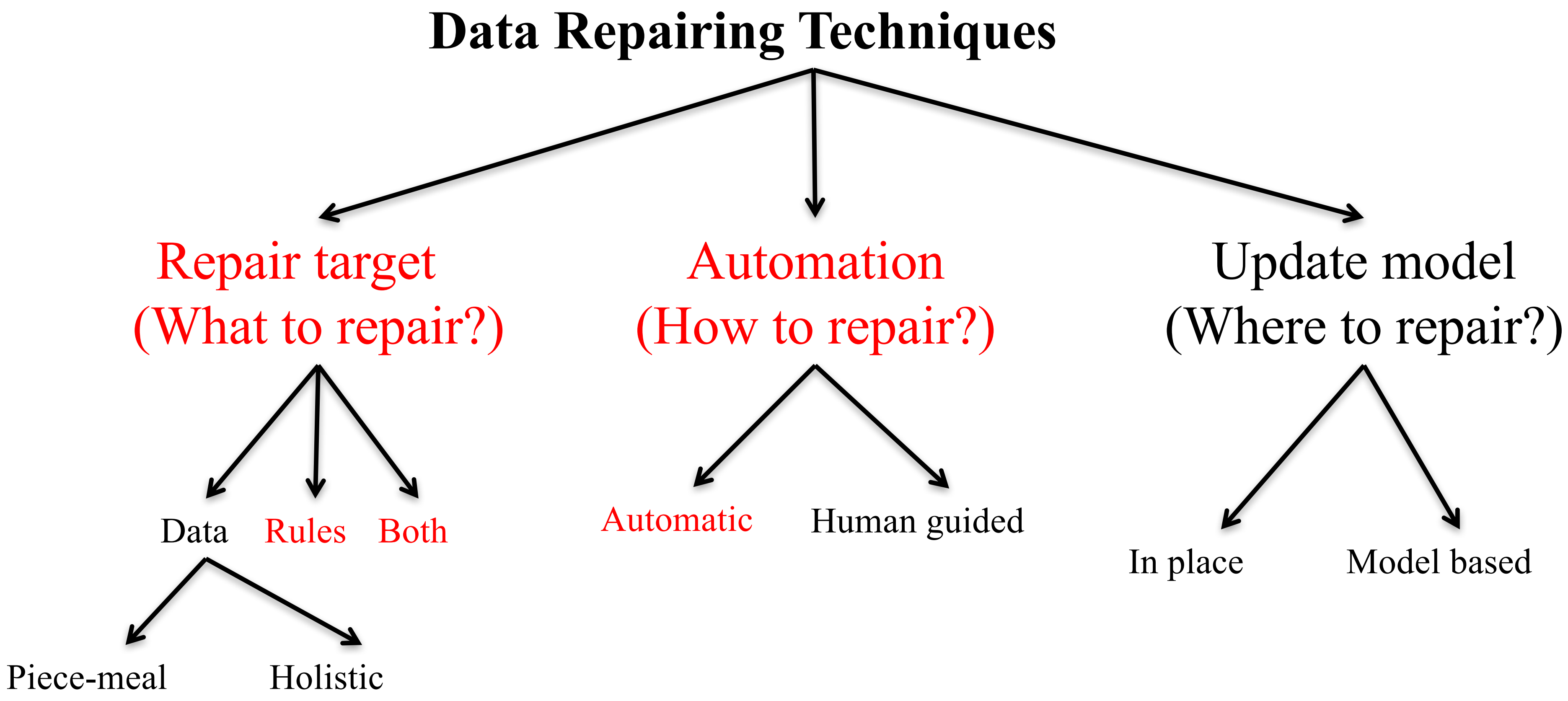
CFD: Tran(AC = 020 → City = Lnd)

CFD: Tran(FN = Bob → FN = Robert)

MD: Tran[LN, City, St, Post] = card[LN, City, St, Zip] ^
Tran[FN] ≈ Card[FN] → Tran[FN, Phn] ⇔ Card[FN, Tel]

FD: Tran(City, Phn → St, AC, Post)

Data Repairing Techniques Taxonomy



Data & Rules Repairing: Motivating Example

- Car Database
  - Model → Make was satisfied by Car databases till Mazda 323 was introduced (Conflicting with BMW 323)
  - Could be corrected to Model, Cylinders→ Make

- US presidents Database
  - LastName, FirstName → StartYear, EndYear was satisfied till the election of George W. Bush
  - Should be corrected to LastName, MiddleInit, FirstName → StartYear, EndYear

[Chiang and Miller, ICDE 2011]

[Beskales et al, ICDE 2013]

Relative Trust

- In reality, both data and constraints (FDs) can be wrong

- The relative trust in data vs. FDs determines how we should repair data and FDs

Example (Trusted FD)

	GivenName	Surname	BirthDate	Gender	Phone	Income
t_1	Danielle	Blake	9 Dec 1970	Female	817-213-1211	120k
t_2	Danielle	Blake	9 Dec 1970	Female	817-988-9211	**120k**
t_3	Hong	Li	27 Oct 1972	Female	591-977-1244	90k
t_4	Hong	Li	8 Mar 1979	Female	498-214-5822	**90k**
t_5	Ning	Wu	3 Nov 1982	Male	313-134-9241	**95k**
t_6	Ning	Wu	8 Nov 1982	Male	323-456-3452	95k

Surname, GivenName → Income

Example (Trusted Data)

	GivenName	Surname	BirthDate	Gender	Phone	Income
t_1	Danielle	Blake	9 Dec 1970	Female	817-213-1211	120k
t_2	Danielle	Blake	9 Dec 1970	Female	817-988-9211	100k
t_3	Hong	Li	27 Oct 1972	Female	591-977-1244	90k
t_4	Hong	Li	8 Mar 1979	Female	498-214-5822	84k
t_5	Ning	Wu	3 Nov 1982	Male	313-134-9241	90k
t_6	Ning	Wu	8 Nov 1982	Male	323-456-3452	95k

Surname, GivenName → Income

Surname, GivenName, BirthDate, Phone → Income

Example (Equally-trusted Data and FD)

	GivenName	Surname	BirthDate	Gender	Phone	Income
t_1	Danielle	Blake	9 Dec 1970	Female	817-213-1211	120k
t_2	Danielle	Blake	9 Dec 1970	Female	817-988-9211	100k **120k**
t_3	Hong	Li	27 Oct 1972	Female	591-977-1244	90k
t_4	Hong	Li	8 Mar 1979	Female	498-214-5822	84k
t_5	Ning	Wu	3 Nov 1982	Male	313-134-9241	90k
t_6	Ning	Wu	8 Nov 1982	Male	323-456-3452	95k

Surname, GivenName → Income

Surname, GivenName, BirthDate → Income

Data Repair

- We repair instance I by modifying multiple cells and produce I'
- $\mathrm{dist}_d(I,I')$ is the number of different cells between I and I'

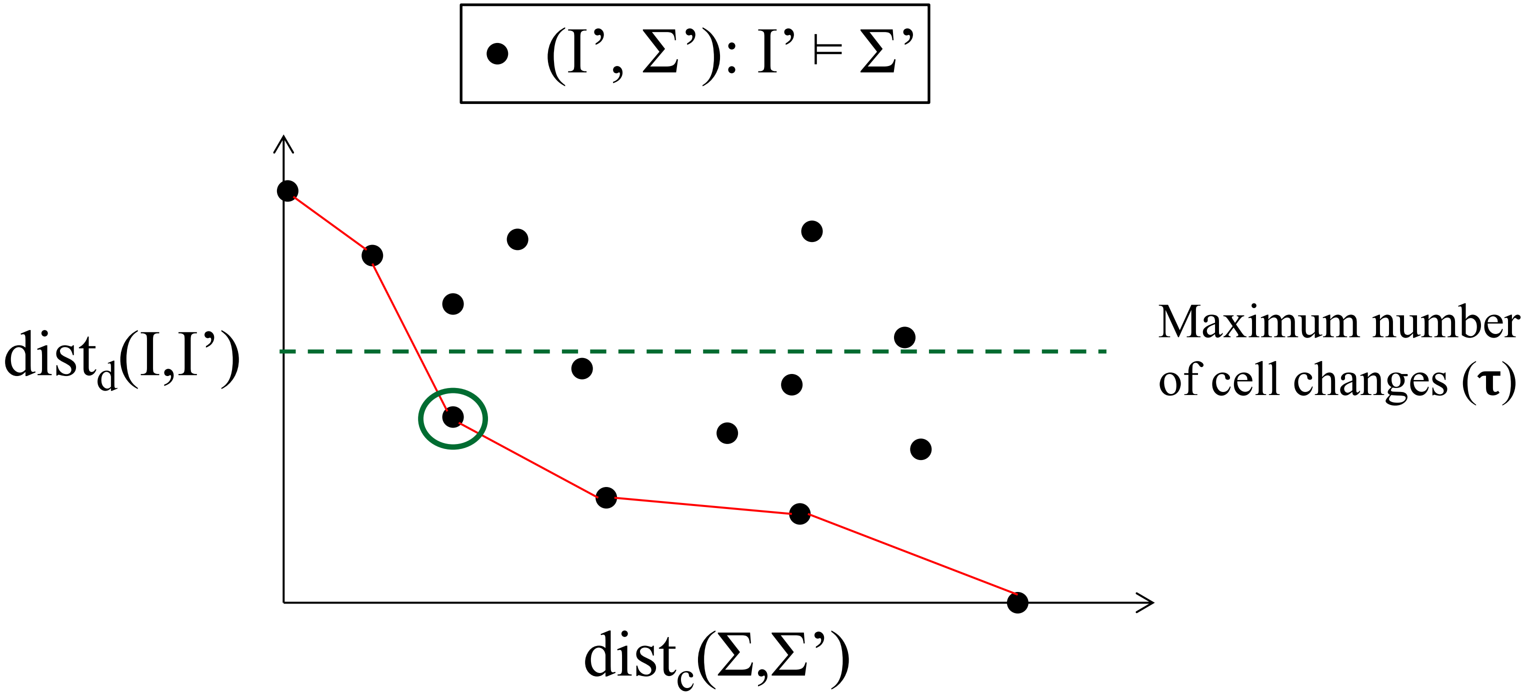
Repairing a set of FDs

- We repair an FD X→A by adding one or more attributes to the LHS

- Let w(Y) be a weight reflecting the penalty of adding attribute set Y to X
  - E.g., the number of attributes in Y, distinct values of Y in I, entropy of Y

- Let $dist_c(\Sigma,\Sigma')$ be the sum of w(Y) across all changed FDs

Relative Trust [Beskales et al, ICDE 2013]



A Unified Cost Model [Chiang and Miller, ICDE 2011]

- Minimum description Length Principle
  - Find a model M w.r.t. Σ that can represent the data as much as possible

- DL(M) = L(M) + L(I|M)
  - L(M): Length of the model
  - L(I|M): Length of data given M

A Unified Cost Model: Data Repair

- M: empty
  - L(M) = 0
  - L(I|M) = 27
  - DL = 27
- M:

Brook	Granville	412

  - L(M) = 3+2*6 = 15
  - L(I|M) = 0
  - DL = 15

FD: {District, Region → AC}

District	Region	AC
Brook	Granville	412
Brook	Granville	412
Brook	Granville	412
Brook	Granville	~~553~~412
Brook	Granville	~~553~~412
Brook	Granville	~~553~~412
Brook	Granville	~~725~~412
Brook	Granville	~~725~~412
Brook	Granville	~~725~~412

A Unified Cost Model: FD Repair

- M: empty
  - L(M) = 0
  - L(I|M) = 36
  - DL = 36
- M:

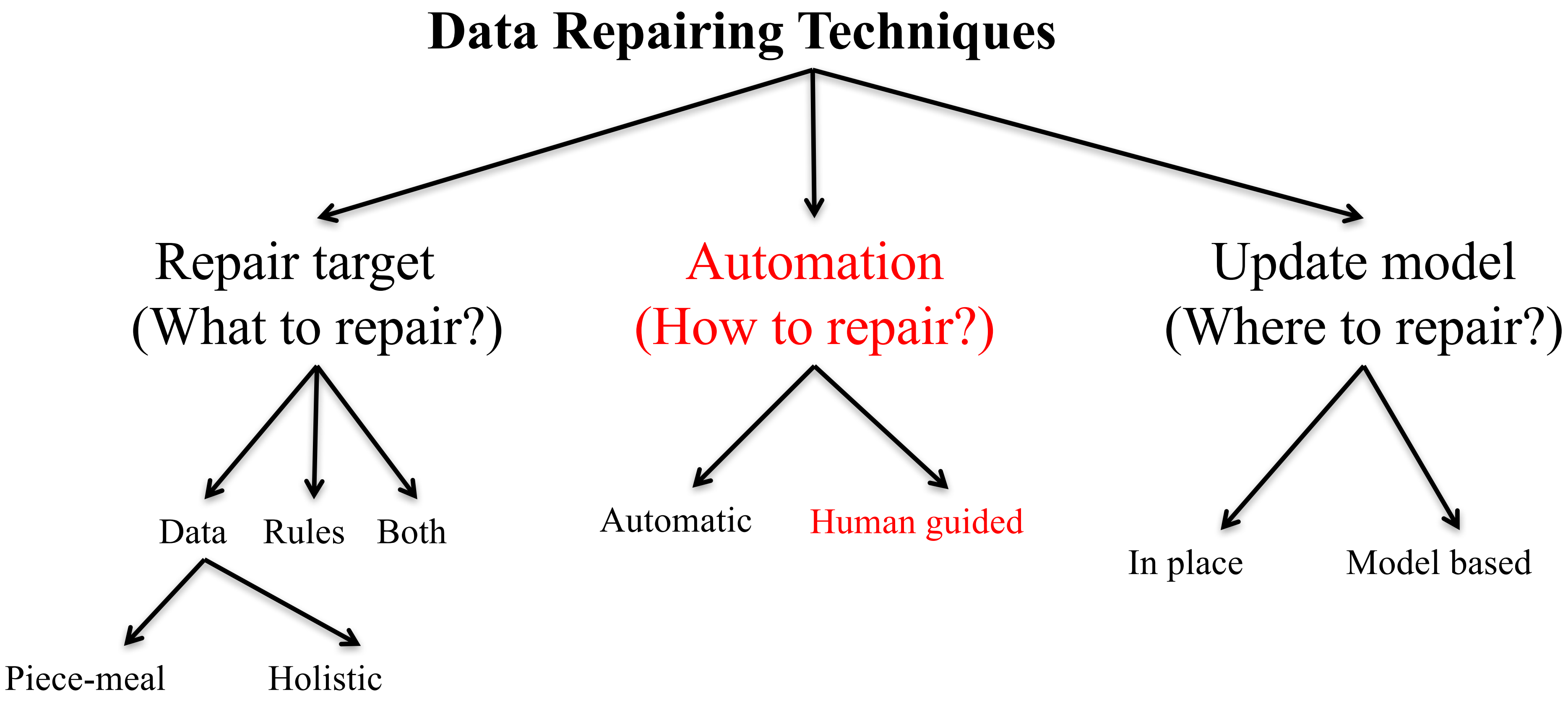
Glendale	Brook	Granville	412
Guildwood	Brook	Granville	553
Moore	Brook	Granville	725

  - L(M) = 12
  - L(I|M) = 0
  - DL = 12

FD: {Municipal, District, Region → AC}

Municipal	District	Region	AC
Glendale	Brook	Granville	412
Glendale	Brook	Granville	412
Glendale	Brook	Granville	412
Guildwood	Brook	Granville	553
Guildwood	Brook	Granville	553
Guildwood	Brook	Granville	553
Moore	Brook	Granville	725
Moore	Brook	Granville	725
Moore	Brook	Granville	725

Data Repairing Techniques Taxonomy



Data Repairing Techniques

Repair target
(What to repair?)

Automation
(How to repair?)

Update model
(Where to repair?)

Data

Rules

Both

Automatic

Human guided

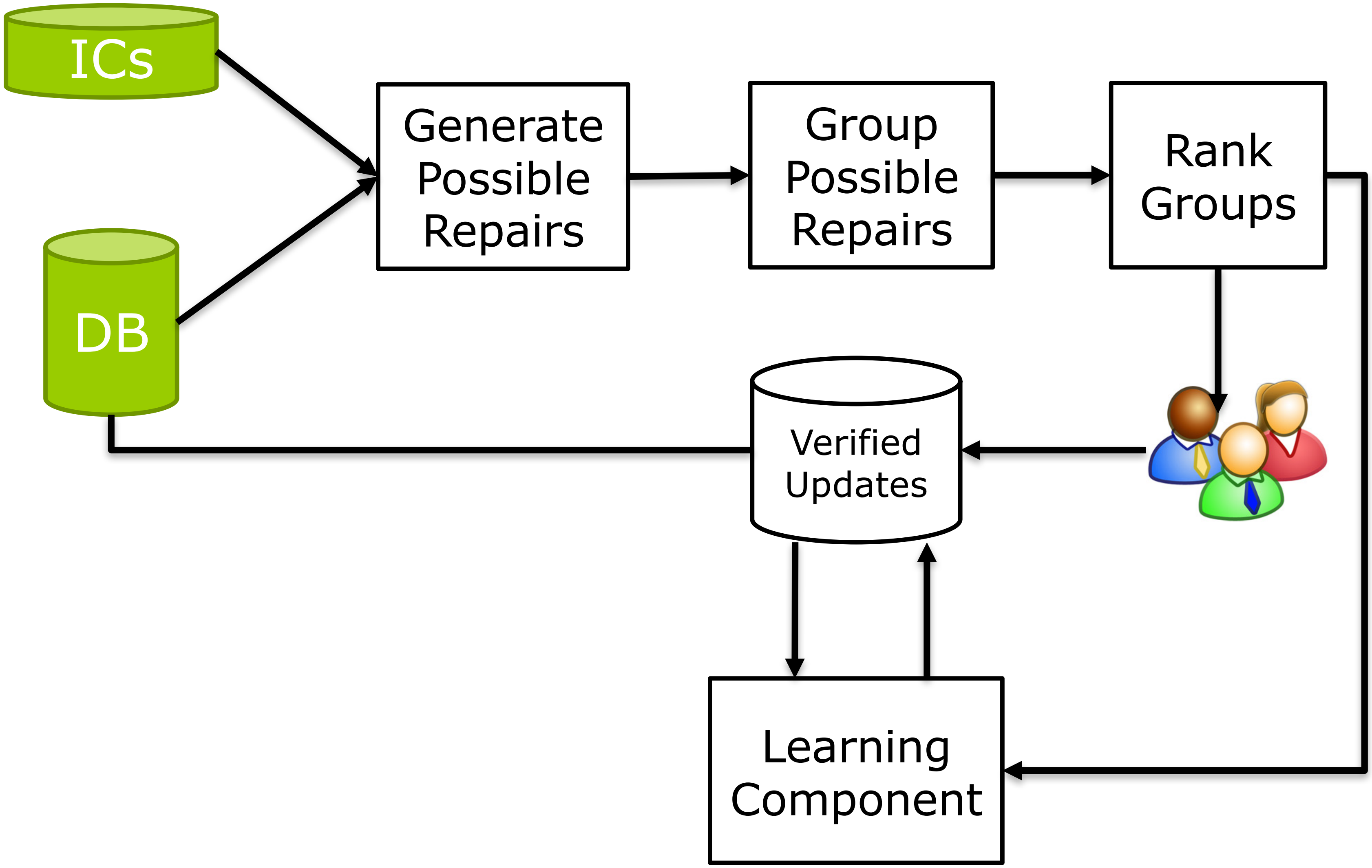
In place

Model based

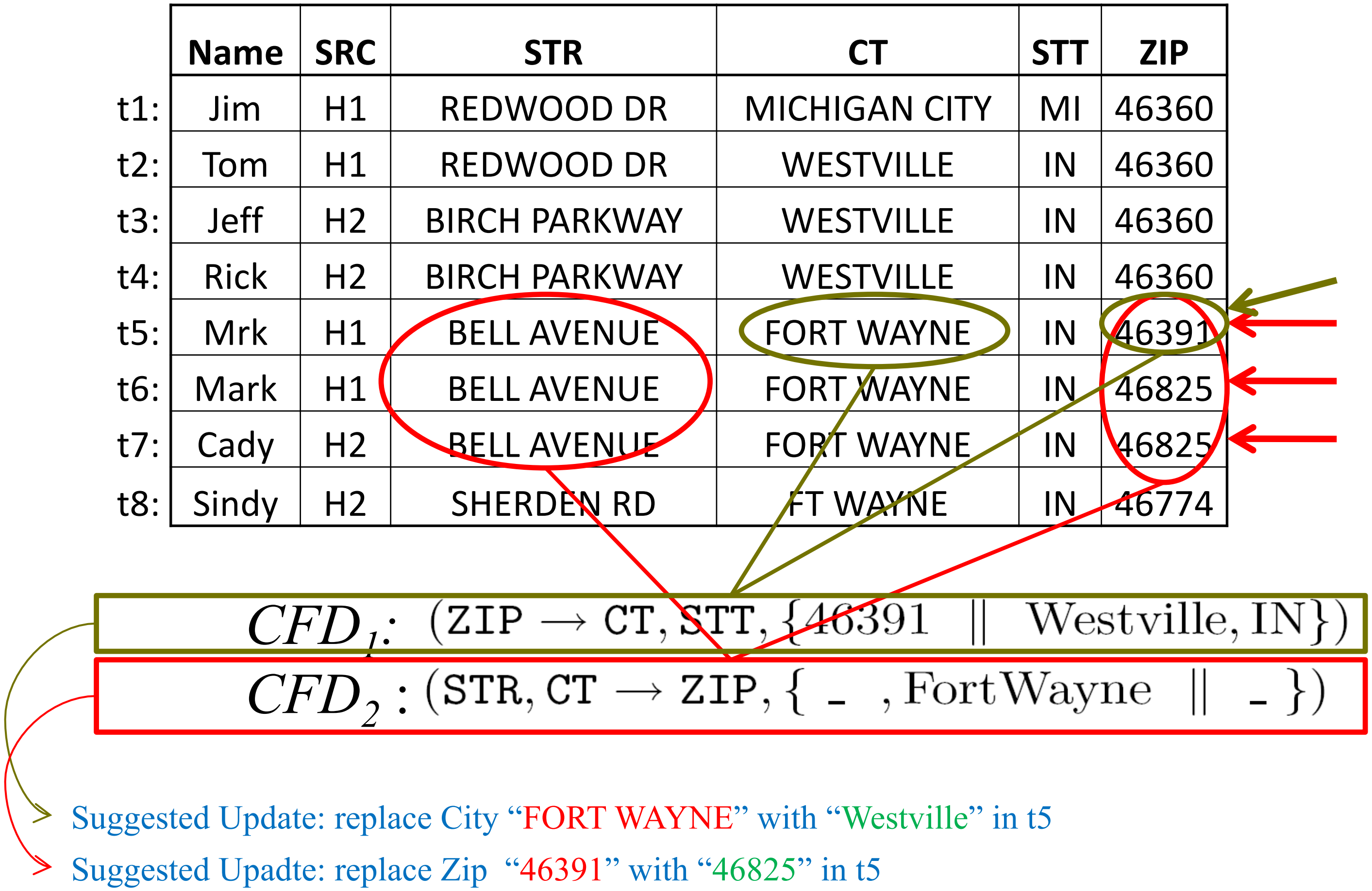
Piece-meal

Holistic

Guided Data Repair (GDR) [Yakout et al, VLDB 2011]



GDR: Generate Possible Repairs



	Name	SRC	STR	CT	STT	ZIP
t1:	Jim	H1	REDWOOD DR	MICHIGAN CITY	MI	46360
t2:	Tom	H1	REDWOOD DR	WESTVILLE	IN	46360
t3:	Jeff	H2	BIRCH PARKWAY	WESTVILLE	IN	46360
t4:	Rick	H2	BIRCH PARKWAY	WESTVILLE	IN	46360
t5:	Mrk	H1	BELL AVENUE	FORT WAYNE	IN	46391
t6:	Mark	H1	BELL AVENUE	FORT WAYNE	IN	46825
t7:	Cady	H2	BELL AVENUE	FORT WAYNE	IN	46825
t8:	Sindy	H2	SHERDEN RD	FT WAYNE	IN	46774

$$CFD_1: (\mathtt{ZIP} \rightarrow \mathtt{CT}, \mathtt{STT}, \{46391 \ \| \ \text{Westville}, \text{IN}\})$$

$$CFD_2: (\mathtt{STR}, \mathtt{CT} \rightarrow \mathtt{ZIP}, \{\ \_\ , \text{Fort Wayne} \ \| \ \_\ \})$$

Suggested Update: replace City “FORT WAYNE” with “Westville” in t5

Suggested Upadte: replace Zip “46391” with “46825” in t5

GDR: Group and Rank Repairs

	Name	SRC	STR	CT	STT	ZIP
t1:	Jim	H1	REDWOOD DR	MICHIGAN CITY	MI	46360
t2:	Tom	H1	REDWOOD DR	WESTVILLE	IN	46360
t3:	Jeff	H2	BIRCH PARKWAY	WESTVILLE	IN	46360
t4:	Rick	H2	BIRCH PARKWAY	WESTVILLE	IN	46360
t5:	Mrk	H1	BELL AVENUE	FORT WAYNE	IN	46391
t6:	Mark	H1	BELL AVENUE	FORT WAYNE	IN	46825
t7:	Cady	H2	BELL AVENUE	FORT WAYNE	IN	46825
t8:	Sindy	H2	SHERDEN RD	FT WAYNE	IN	46774

Contextual grouping for the suggested updates

Update Group g_1: The city should be “Michigan City” for $\{t_2, t_3, t_4\}$.
Update Group g_2: The zip should be “46825” for $\{t_5, t_8\}$.
….
….
….

KATARA [Chu et al, SIGMOD 2015]

	A	B	C	D	E	F	G
t_1	Rossi	Italy	Rome	Verona	Italian	Proto	1.78
t_2	Klate	South Africa	Pretoria	Pirates	Afrikaans	P. Eliz.	1.69
t_3	Pirlo	Italy	Madrid	Juve	Italian	Flero	1.77

A Table of Soccer Players

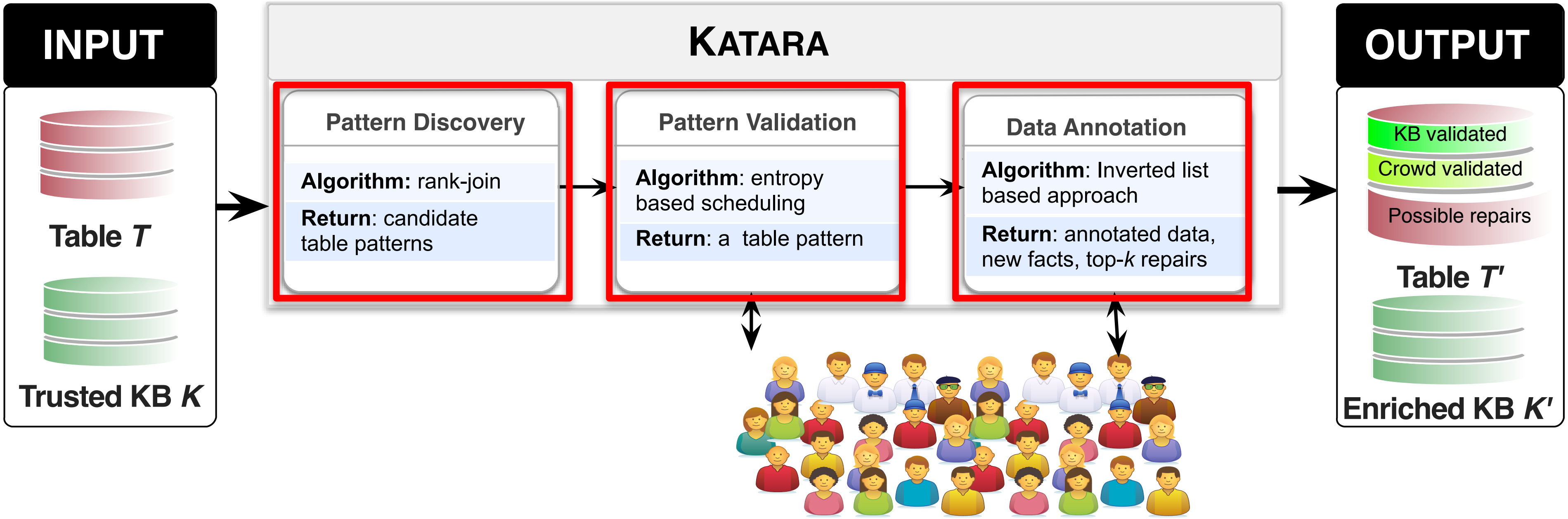
FD: $B \rightarrow C$

- Automatic: Produce heuristic repairs
- GDR:
  - Rely on redundancy to detect errors
  - Require heavy human involvement

Proposal: Use external trustworthy information!

- KBs
- Crowd experts

KATARA Workflow



INPUT

Table *T*

Trusted KB *K*

KATARA

Pattern Discovery

Algorithm: rank-join

Return: candidate table patterns

Pattern Validation

Algorithm: entropy based scheduling

Return: a table pattern

Data Annotation

Algorithm: Inverted list based approach

Return: annotated data, new facts, top-*k* repairs

OUTPUT

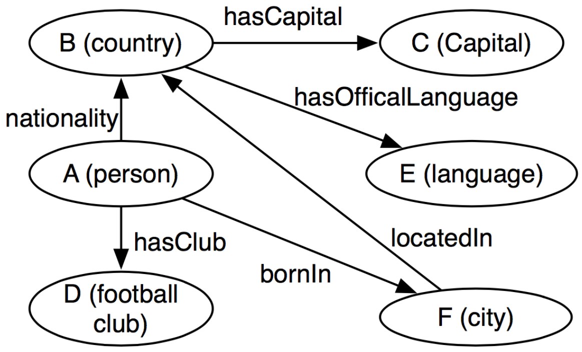
KB validated

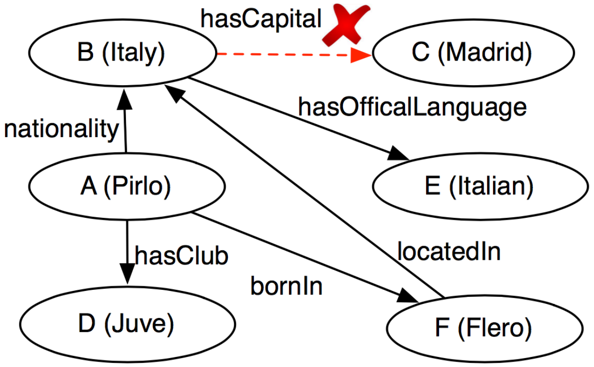
Crowd validated

Possible repairs

Table *T′*

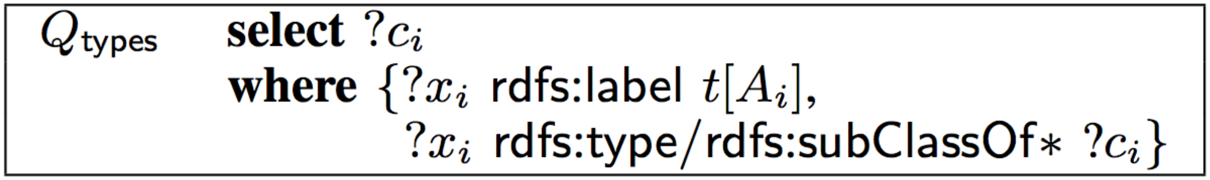
Enriched KB *K′*





Pattern Discovery: Generate Candidates

Generate candidate types for every column:



Q_{types} **select** $?c_i$
where $\{?x_i$ rdfs:label $t[A_i]$,
$?x_i$ rdfs:type/rdfs:subClassOf$*$ $?c_i\}$

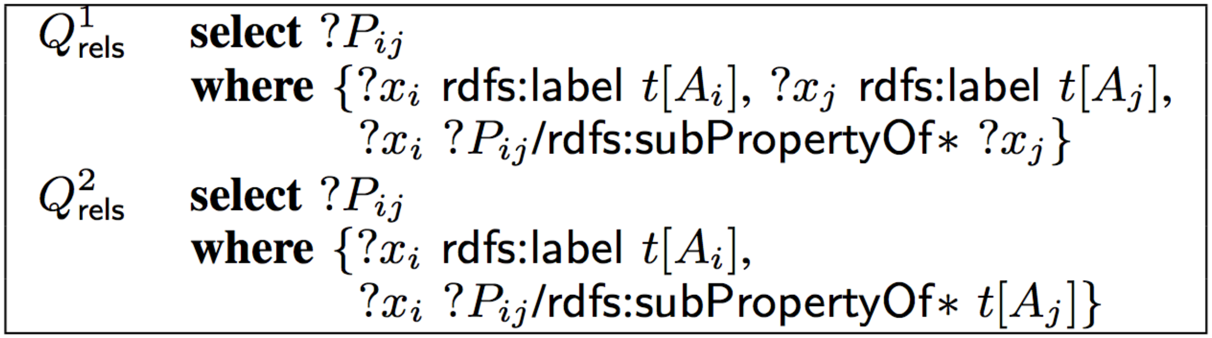
type(*B*)

economy
country
location
state
…

type (*C*)

City
Capital
whole
artifact
Person
…

Generate candidate relationships for every column pair:



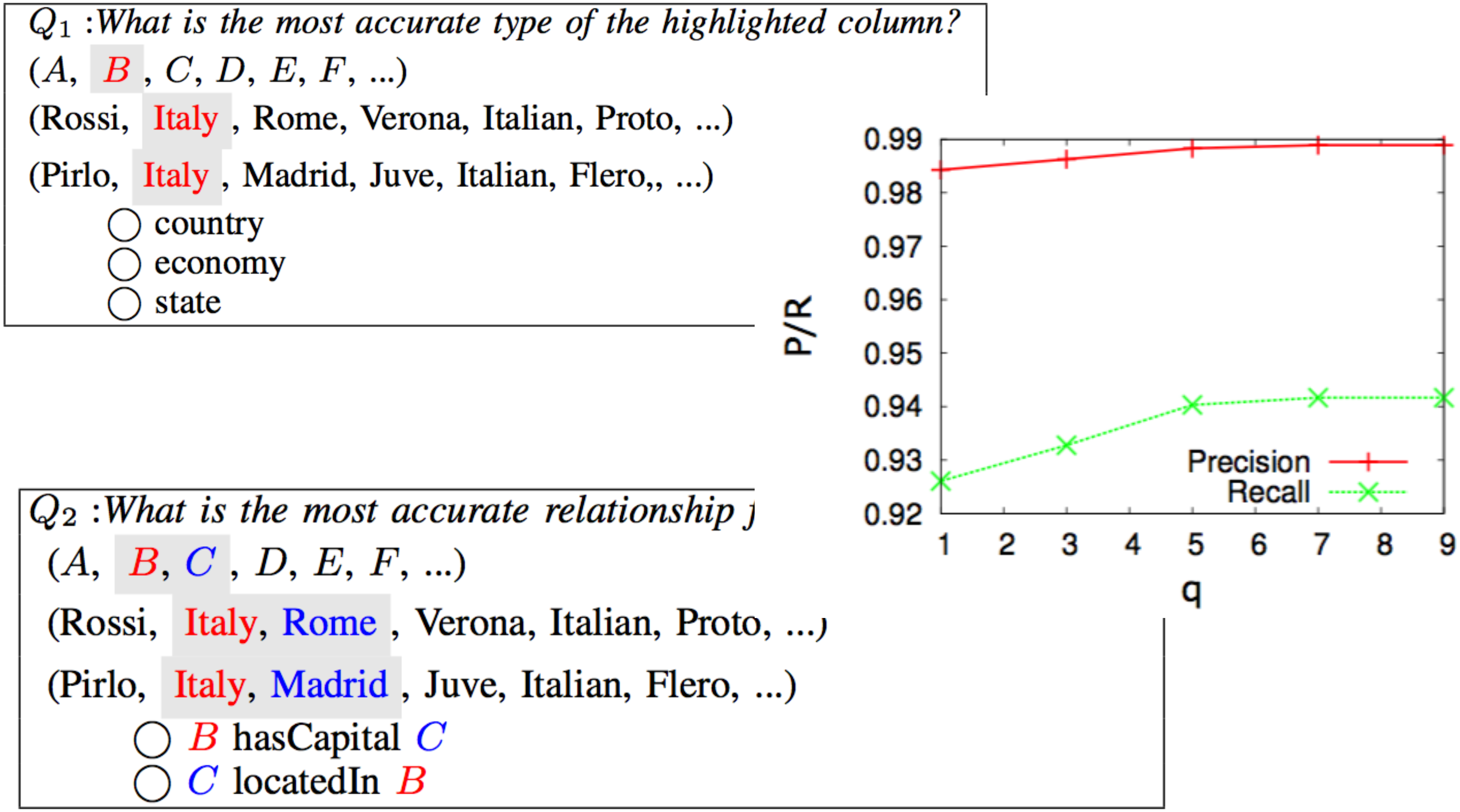
Q^1_{rels} **select** $?P_{ij}$
where $\{?x_i$ rdfs:label $t[A_i]$, $?x_j$ rdfs:label $t[A_j]$,
$?x_i$ $?P_{ij}$/rdfs:subPropertyOf$*$ $?x_j\}$

Q^2_{rels} **select** $?P_{ij}$
where $\{?x_i$ rdfs:label $t[A_i]$,
$?x_i$ $?P_{ij}$/rdfs:subPropertyOf$*$ $t[A_j]\}$

relationship (*B*, *C*)

locatedIn
hasCapital

Crowd Pattern Validation



Q_1 :*What is the most accurate type of the highlighted column?*

(A, B, C, D, E, F, ...)

(Rossi, Italy, Rome, Verona, Italian, Proto, ...)

(Pirlo, Italy, Madrid, Juve, Italian, Flero,, ...)

○ country
○ economy
○ state

Q_2 :*What is the most accurate relationship*

(A, B, C, D, E, F, ...)

(Rossi, Italy, Rome, Verona, Italian, Proto, ...)

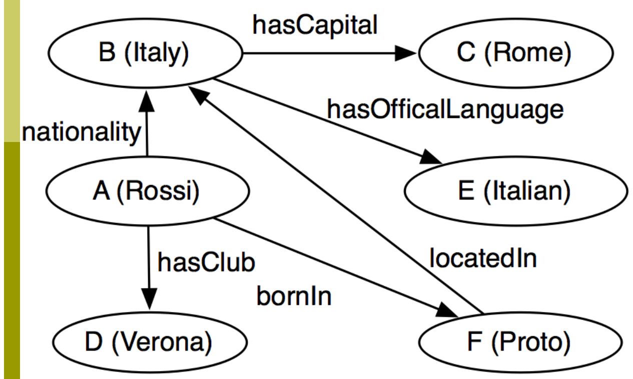
(Pirlo, Italy, Madrid, Juve, Italian, Flero, ...)

○ B hasCapital C
○ C locatedIn B

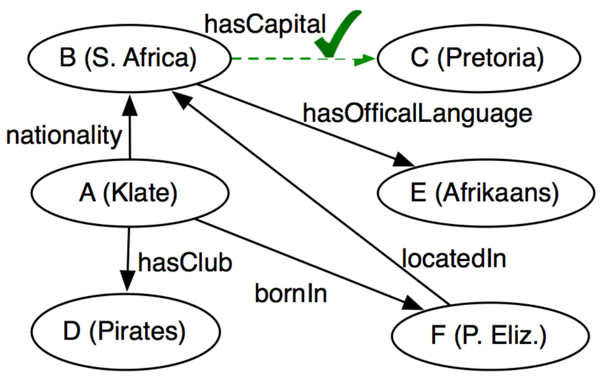
Data Annotation

	A	B	C	D	E	F	G
t_1	Rossi	Italy	Rome	Verona	Italian	Proto	1.78
t_2	Klate	South Africa	Pretoria	Pirates	Afrikaans	P. Eliz.	1.69
t_3	Pirlo	Italy	Madrid	Juve	Italian	Flero	1.77

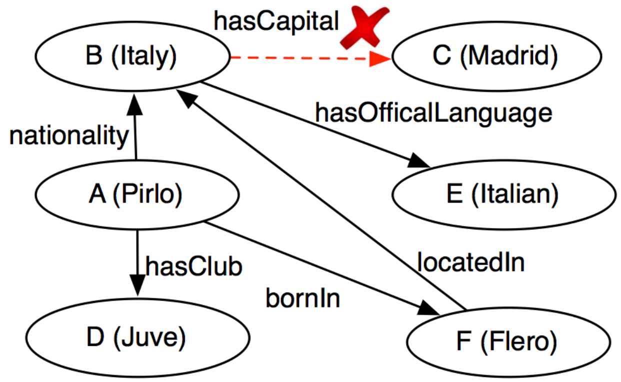
t_1: validated by KB



t_2: validated by KB &crowd



t_3: Erroneous tuple

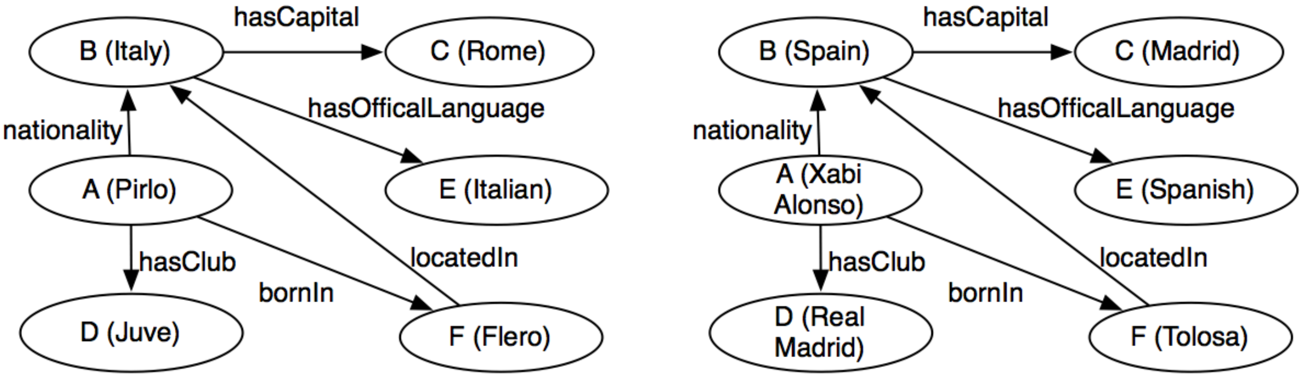


Data Repairing

	A	B	C	D	E	F	G
t_1	Rossi	Italy	Rome	Verona	Italian	Proto	1.78
t_2	Klate	South Africa	Pretoria	Pirates	Afrikaans	P. Eliz.	1.69
t_3	Pirlo	Italy	Madrid	Juve	Italian	Flero	1.77

G_1 has cost 1

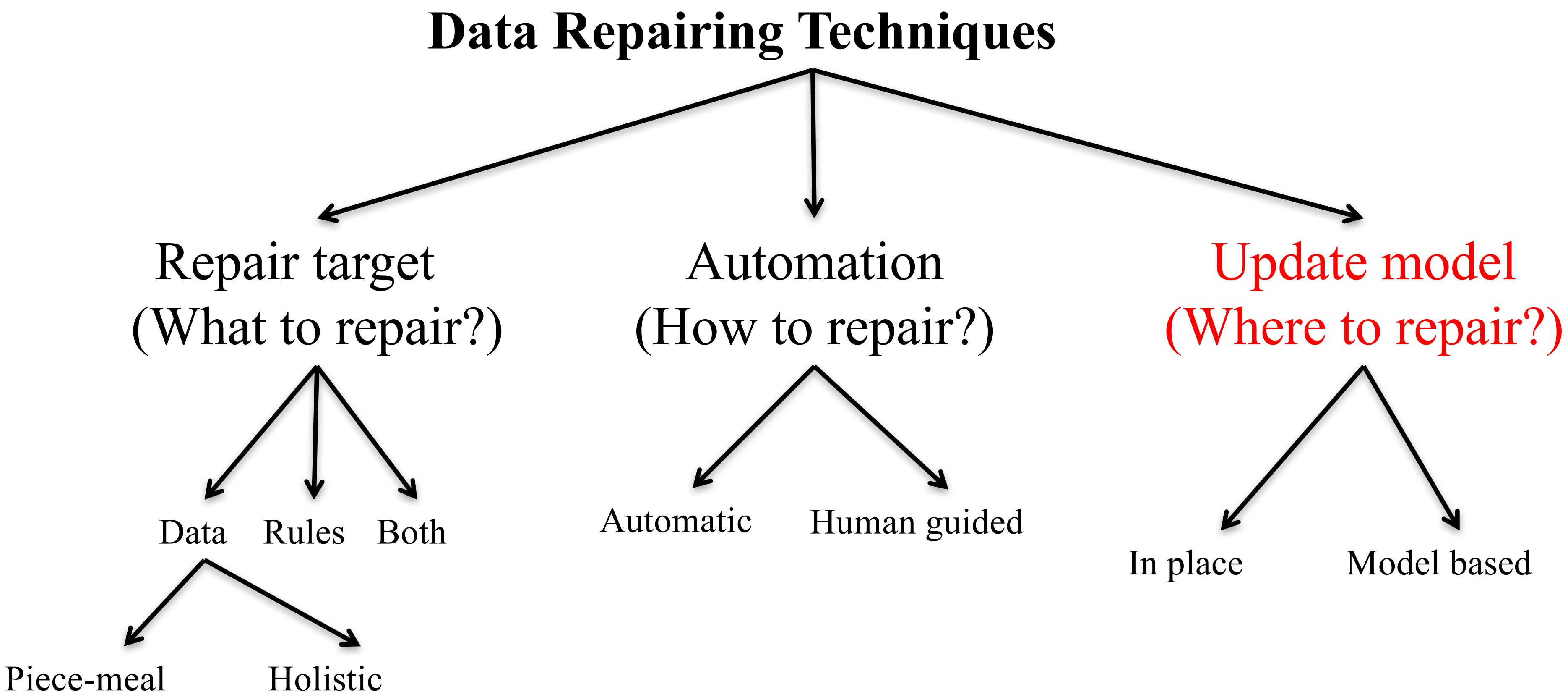
G_2 has cost 5



(a) Possible repair G_1

(b) Possible repair G_2

Data Repairing Techniques Taxonomy



Data Repairing Techniques

Repair target (What to repair?)

Automation (How to repair?)

Update model (Where to repair?)

Data

Rules

Both

Automatic

Human guided

In place

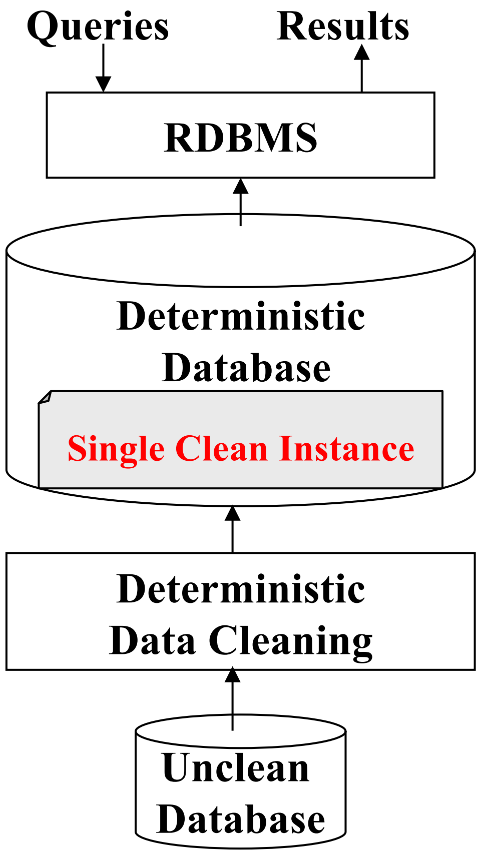
Model based

Piece-meal

Holistic

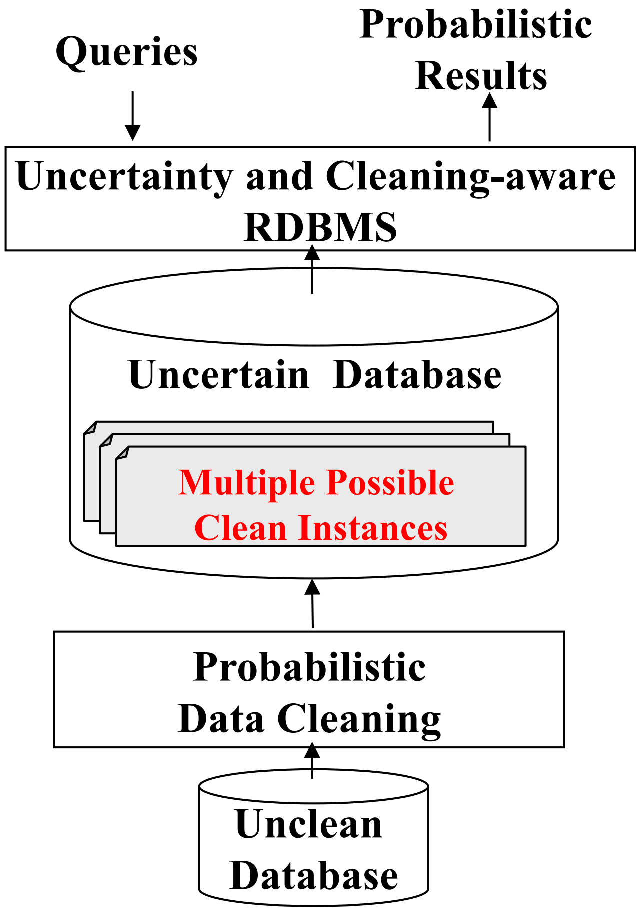
One-Shot Data Cleaning

- Generate a single “trustworthy” instance



Model Based Approach

- Generate multiple possible clean instances


Queries
Probabilistic Results
Uncertainty and Cleaning-aware RDBMS
Uncertain Database
Multiple Possible Clean Instances
Probabilistic Data Cleaning
Unclean Database

Model Based Approach Challenges

1. The space of all possible repairs is huge
2. How to efficiently generate, store and query the possible repairs

Two Example Model Based Approaches

- Duplicate Detection [Beskales et al, VLDB 2009]
  - Spaces of Possible Repairs
  - Generating and Storing Possible Repairs
  - Query Possible Repairs

- Violations of Functional Dependencies [Beskales et al, VLDB 2010]
  - Spaces of Possible Repairs
  - Sampling from a Meaningful Space of Repairs

Two Example Model Based Approaches

- Duplicate Detection [Beskales et al, VLDB 2009]
  - Spaces of Possible Repairs
  - Generating and Storing Possible Repairs
  - Query Possible Repairs

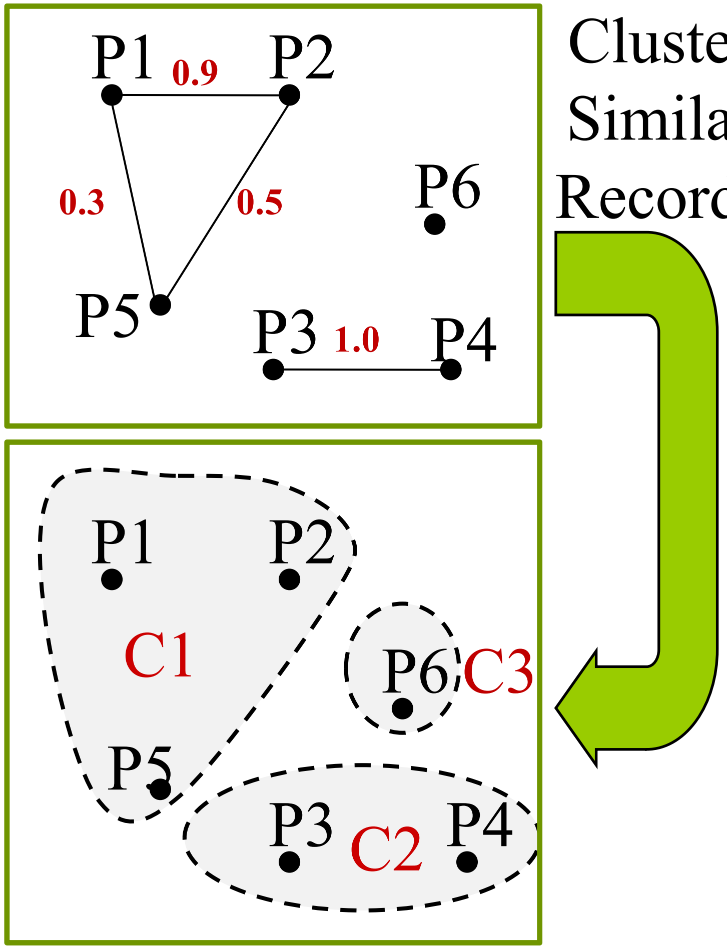
- Violations of Functional Dependencies [Beskales et al, VLDB 2010]
  - Spaces of Possible Repairs
  - Sampling from a Meaningful Space of Repairs

Typical Duplicate Elimination

Unclean Relation

ID	name	ZIP	Income
P1	Green	51519	30k
P2	Green	51518	32k
P3	Peter	30528	40k
P4	Peter	30528	40k
P5	Gree	51519	55k
P6	Chuck	51519	30k

Compute Pair-wise Similarity



Cluster Similar Records

Merge Clusters

Clean Relation

ID	name	ZIP	Income
C1	Green	51519	39k
C2	Peter	30528	40k
C3	Chuck	51519	30k

Possible Repairs [Beskales et al, VLDB 2009]

- A **possible repair** is a clustering (partitioning) of the input tuples

Person

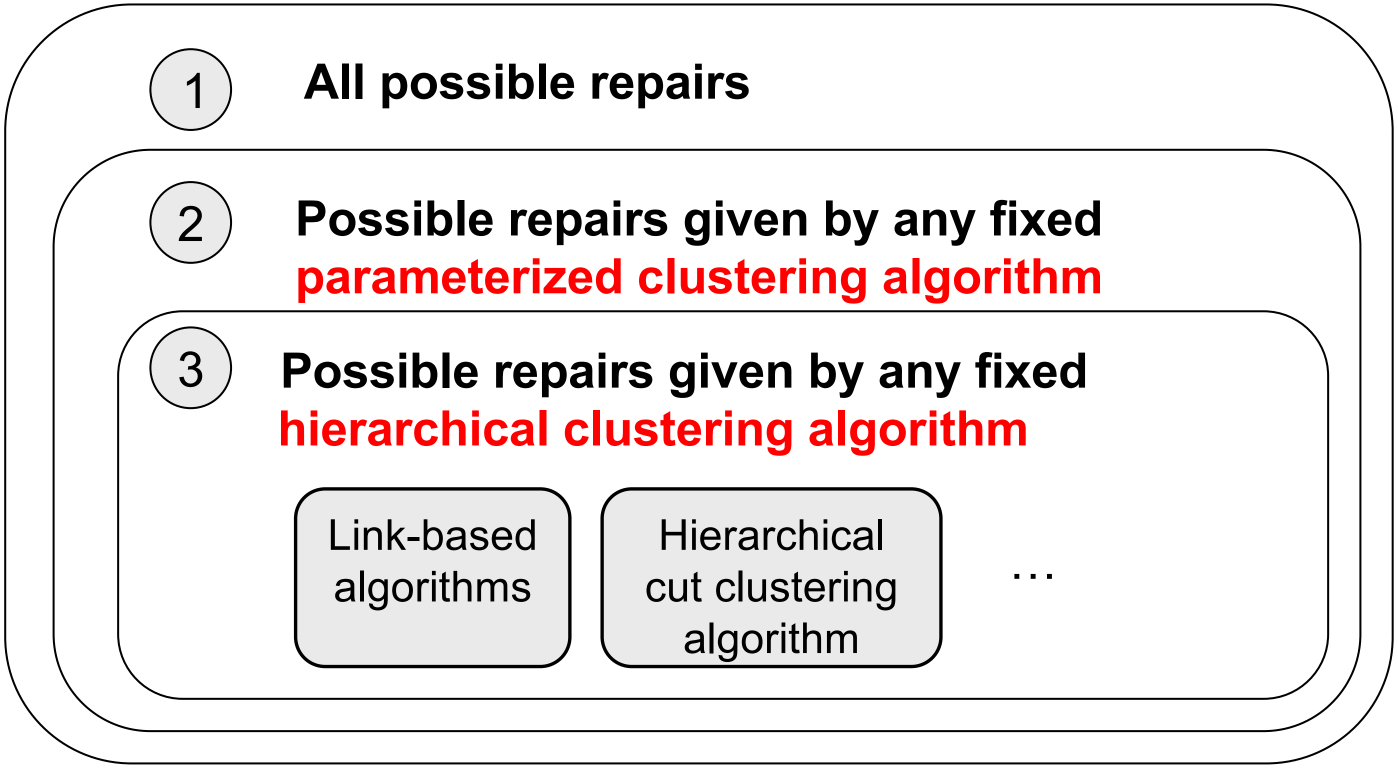
ID	Name	ZIP	Income
P1	Green	51519	30k
P2	Green	51518	32k
P3	Peter	30528	40k
P4	Peter	30528	40k
P5	Gree	51519	55k
P6	Chuck	51519	30k

Uncertain Clustering

Possible Repairs

X_1	X_2	X_3
{P1}	{P1,P2}	{P1,P2,P5}
{P2}	{P3,P4}	{P3,P4}
{P3,P4}	{P5}	{P6}
{P5}	{P6}	
{P6}		

Spaces of Possible Repairs

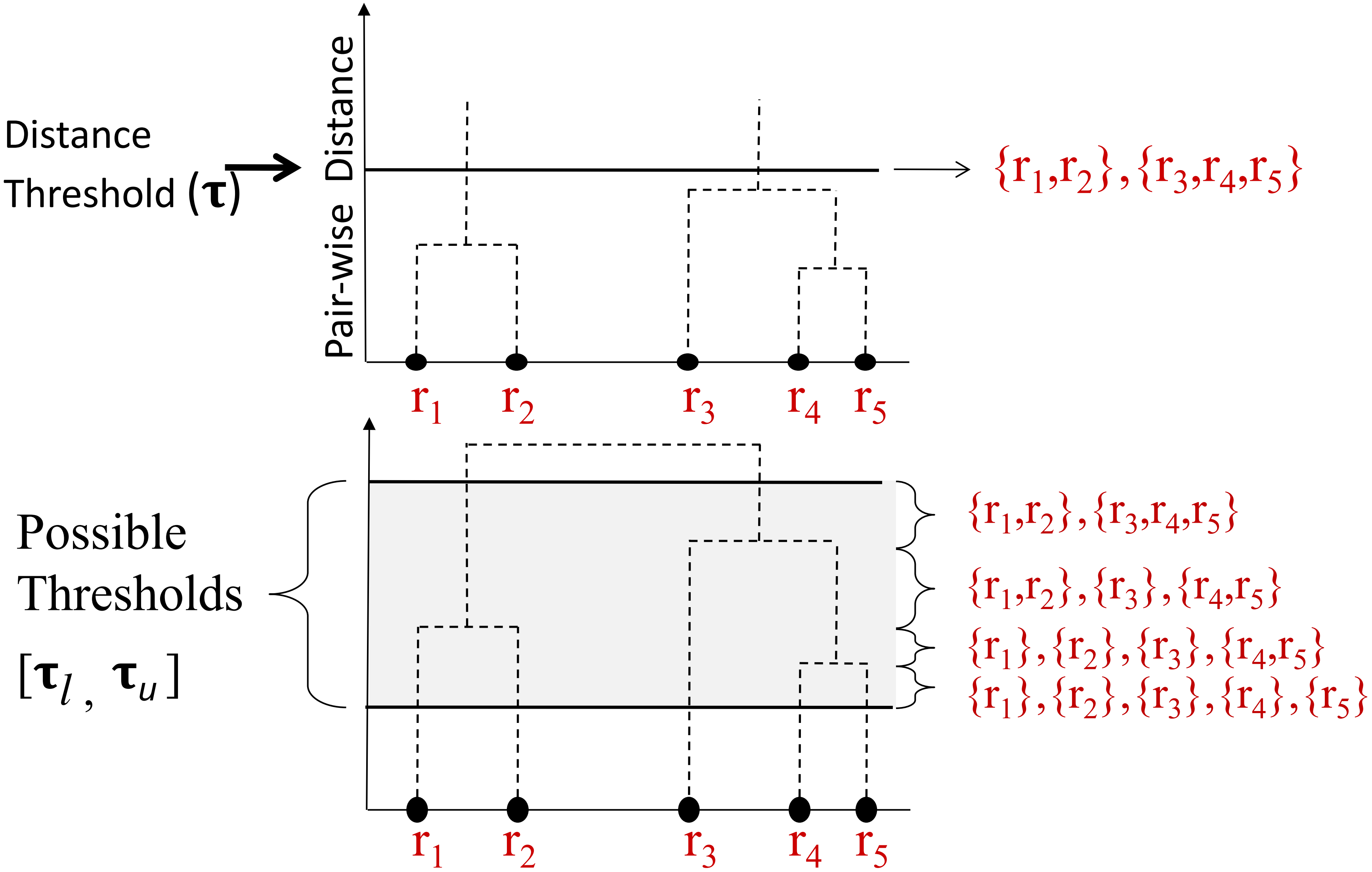


1 All possible repairs

2 Possible repairs given by any fixed **parameterized clustering algorithm**

3 Possible repairs given by any fixed **hierarchical clustering algorithm**

Generating Possible Repairs



Distance Threshold (τ)

Pair-wise Distance

$\{r_1,r_2\},\{r_3,r_4,r_5\}$

r_1 r_2 r_3 r_4 r_5

Possible Thresholds

$[\tau_l, \tau_u]$

$\{r_1,r_2\},\{r_3,r_4,r_5\}$

$\{r_1,r_2\},\{r_3\},\{r_4,r_5\}$

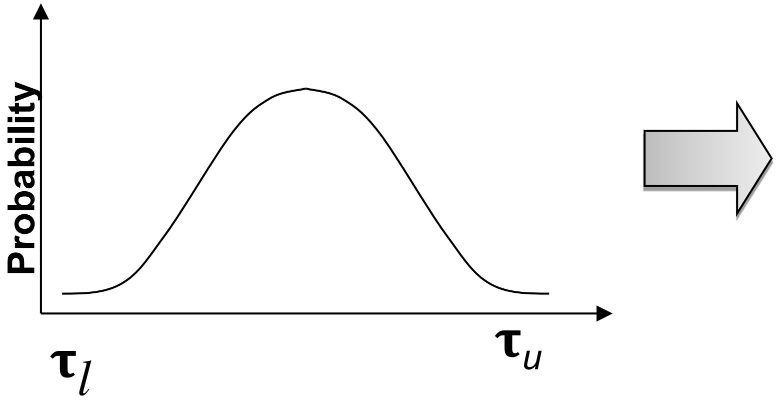
$\{r_1\},\{r_2\},\{r_3\},\{r_4,r_5\}$

$\{r_1\},\{r_2\},\{r_3\},\{r_4\},\{r_5\}$

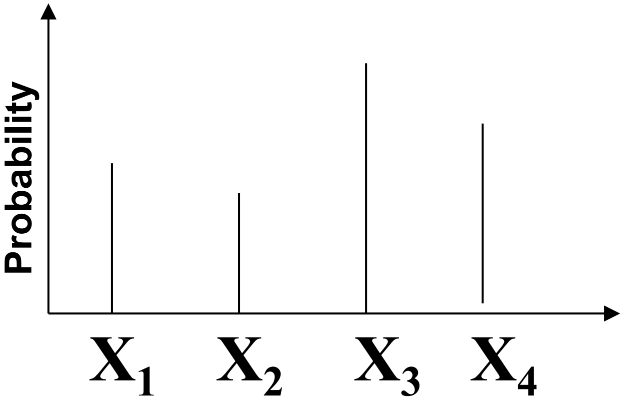
r_1 r_2 r_3 r_4 r_5

Probabilities of Possible Repairs

- The probability of a repair is equal to the probability of the parameter range that generates such repair



Probability Distribution of τ



Probability Distribution of repairs

Storing Possible Repairs

- U-Clean Relations
  - Each cluster is stored once
  - We keep the “lineage” of each cluster

Clustering 1	Clustering 2	Clustering 3
{P1}	{P1,P2}	{P1,P2,P5}
{P2}	{P3,P4}	{P3,P4}
{P3,P4}	{P5}	{P6}
{P5}	{P6}	
{P6}		
$0 \leq \boldsymbol{\tau}<1$	$1\leq \boldsymbol{\tau}<3$	$3 \leq \boldsymbol{\tau}<10$

U-clean Relation *Person*^{C}

ID	…	Income	C	P
CP1	…	31k	{P1,P2}	[1,3)
CP2	**…**	**40k**	**{P3,P4}**	**[0,10)**
CP3	**…**	**55k**	**{P5}**	**[0,3)**
CP4	…	30k	{P6}	[0,10)
CP5	**…**	**39k**	**{P1,P2,P5}**	**[3,10)**
CP6	…	30k	{P1}	[0,1)
CP7	…	32k	{P2}	[0,1)

Example: Projection Query

Person^{C}

ID	…	Income	C	P
CP1	…	31k	{P1,P2}	[1,3)
CP2	…	40k	{P3,P4}	[0,1)
CP3	…	55k	{P5}	[0,3)
CP4	…	30k	{P6}	[3,10)
CP5	…	40k	{P1,P2,P5}	[3,10)
CP6	…	30k	{P1}	[0,1)
CP7	…	32k	{P2}	[0,1)

SELECT DISTINCT Income
FROM Person^{c}

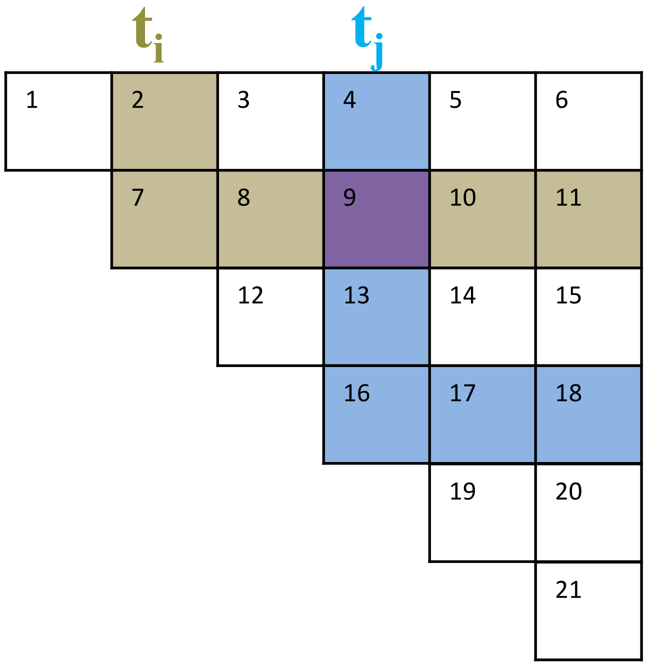
Income	C	P
30k	{P1} v {P6}	[0,1) v [3,10)
31k	{P1,P2}	[1,3)
32k	{P2}	[0,1)
40k	{P3,P4} v {P1,P2,P5}	[0,1) v [3,10)
55k	{P5}	[0,3)

Big Data Cleaning Challenges

- Volume
  - Distributed Data Cleaning
  - Sample Clean (Part 2)
- Velocity
  - Incremental Data Cleaning
- Variety
  - Graph/JSON/RDF
  - Text

Distributed Data Deduplication [Chu et al, VLDB 2016]

- Data deduplication in data lake setting
  - A shared-nothing environment
  - Need to compare every tuple pair
- The goal is to minimizing
  - Largest communication cost
  - Largest computation cost



Conclusion and References

- Error Detection
  - What (IC Languages and Discovery)

- P. Bohannon, W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for data cleaning. In Proceedings of the 23rd International Conference on Data Engineering, pages 746–755, 2007.
- X. Chu, I. F. Ilyas, and P. Papotti. Discovering denial constraints. Proceedings of the VLDB Endowment, 6(13):1498–1509, 2013.
- W. Fan, X. Jia, J. Li, and S. Ma. Reasoning about record matching rules. Proceedings of the VLDB Endowment, 2(1):407–418, 2009.
- N. Koudas, A. Saha, D. Srivastava, and S. Venkatasubramanian. Metric functional dependencies. In Proceedings of the 25th International Conference on Data Engineering, pages 1275–1278, 2009.
- G. Fan, W. Fan, and F. Geerts. Detecting errors in numeric attributes.In Web-Age Information Management, pages 125–137. Springer, 2014.
- W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. Proceedings of the VLDB Endowment, 3(1-2):173–184, 2010.
- J. Wang and N. Tang. Towards dependable data repairing with fixing rules. In Proceedings of the 2014 ACM SIGMOD International Confer- ence on Management of Data, pages 457–468. ACM, 2014.
- M. Interlandi and N. Tang. Proof positive and negative in data cleaning.In 31st IEEE International Conference on Data Engineering, 2015.
- Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. TANE: An efficient algorithm for discovering functional and approximate depen- dencies. Computer Journal, 42(2):100–111, 1999.
- C. M. Wyss, C. Giannella, and E. L. Robertson. FastFDs: A heuristic- driven, depth-first algorithm for mining functional dependencies from relation instances. In International Conference on Big Data Analytics and Knowledge Discovery, pages 101–110, 2001.

Conclusion and References

- Error Detection
  - How (Human involvement)
    - X. Chu, I. F. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In 29th IEEE International Conference on Data Engineering, pages 458–469, 2013.
    - J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. Proceedings of the VLDB Endowment, 5(11):1483– 1494, 2012.
  - Where (Analytics Layer)
    - A. Chalamalla, I. F. Ilyas, M. Ouzzani, and P. Papotti. Descriptive and prescriptive data cleaning. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pages 445–456, 2014.
    - A. Meliou, W. Gatterbauer, S. Nath, and D. Suciu. Tracing data errors with view-conditioned causality. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, pages 505–516, 2011.
    - X. Wang, X Dong, and A. Meliou. Data X-Ray: A Diagnostic Tool for Data Errors . In Proceedings of the 2015 ACM SIGMOD International Conference on Management of data, pages 1231-1245, 2011.
    - M. Bergman, T. Milo, S. Novgorodov, and W Tan. QOCO: A Query Oriented Data Cleaning System with Oracles. Proceedings of the VLDB Endowment, 8(12):1900– 1903, 2015.

Conclusion and References

- Error Repairing
  - What (Data or Data & Rule)
    - P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pages 143–154. ACM, 2005.
    - X. Chu, I. F. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In 29th IEEE International Conference on Data Engineering, pages 458–469, 2013.
    - G. Beskales, I. F. Ilyas, L. Golab, and A. Galiullin. On the relative trust between inconsistent data and inaccurate constraints. In 29th IEEE International Conference on Data Engineering, pages 541–552, 2013.
    - F. Chiang and R. J. Miller. A unified model for data and constraint repair. In Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, pages 446–457, 2011.
  - How (Human Involvement)
    - M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. Proceedings of the VLDB Endowment, 4(5):279– 289, 2011.
    - X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye. KATARA: A data cleaning system powered by knowledge bases and crowdsourcing. In Proceedings of the 2015 ACM SIGMOD Inter- national Conference on Management of Data, pages 1247–1261, 2015.

Conclusion and References

- Error Repairing
  - Where (Model-based)
    - G. Beskales, M. A. Soliman, I. F. Ilyas, and S. Ben-David. Modeling and querying possible repairs in duplicate detection. Proceedings of the VLDB Endowment, pages 598–609, 2009.
    - G. Beskales, I. F. Ilyas, and L. Golab. Sampling the repairs of functional dependency violations under hard constraints. Proceedings of the VLDB Endowment, 3(1-2):197–207, 2010.

- Taxonomy
  - I. F. Ilyas, and X. Chu. Trends in Cleaning Relational Data: Consistency and Deduplication . In Foundations and Trends® in Databases, Volume 5, Issue 4, 2015