

---

# **Data Auditor: Analyzing Data Quality Using Pattern Tableaux**

Lukasz Golab, Howard Karloff, Flip Korn,  
Divesh Srivastava  
AT&T Labs-Research

# Key Takeaways

---

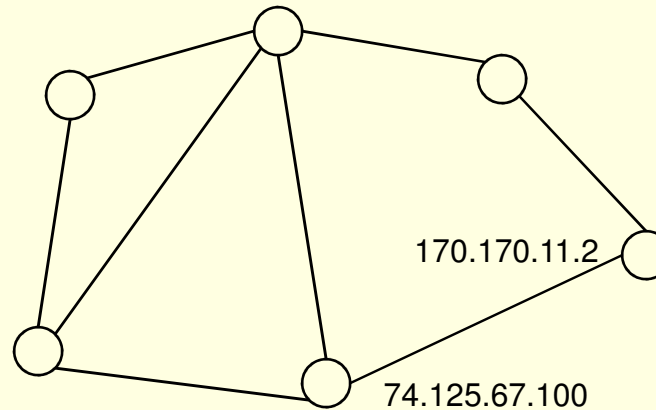
- Philosophy: constraints used to represent database semantics
  - Conditional constraints allow more expressive power
  - Data quality problem: a violation of database semantics
- Contributions: discovery of conditional constraints, tableaux
  - What is an optimal tableau?
  - How difficult is it to find an optimal tableau?
  - How practical and scalable is discovery of a good tableau?

# Outline

---

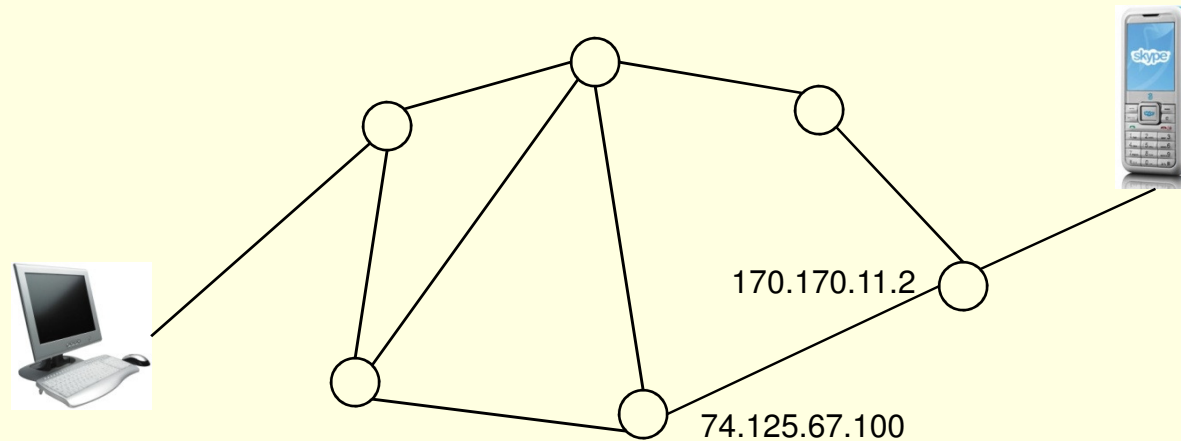
- Monitoring databases: what? how? why?
- Data audits: missing polls, inconsistencies
- Conditional constraints: tableaux generation

# Monitoring Databases: What?



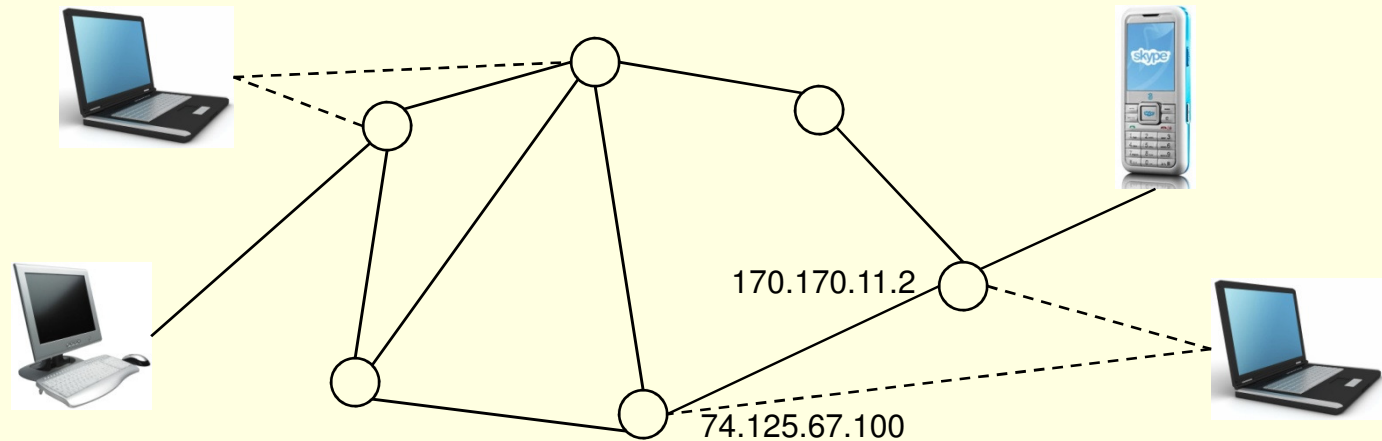
- Configuration tables: keep track of configuration, topology
  - router\_config(timestamp, router, interface, ip\_address, type)
  - router\_topology(timestamp, router1, interf1, router2, interf2)
- How big: several GB for a large network

# Monitoring Databases: What?



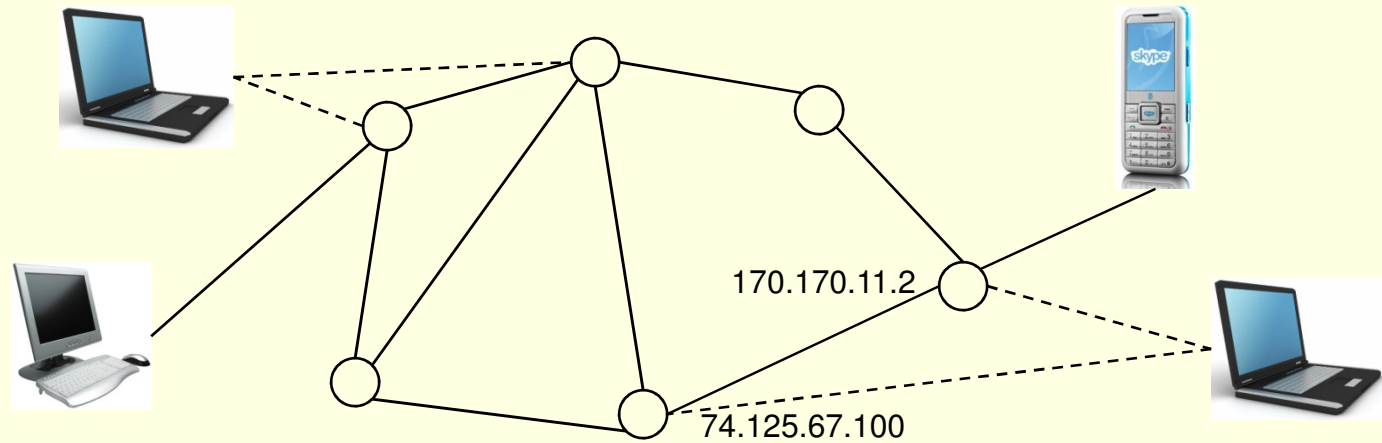
- Measurement tables: keep track of aggregate traffic, usage
  - router\_bps(timestamp, router, interface, in\_traffic, out\_traffic)
  - router\_cpu(timestamp, router, usage)
  
- How big: TB/day for a large network

# Monitoring Databases: How?



- Measurement tables: keep track of aggregate traffic, usage
  - `router_bps(timestamp, router, interface, in_traffic, out_traffic)`
- Pollers use SNMP to periodically poll router interfaces for traffic
  - Poll each interface every 5 minutes = 288 polls/day

# Monitoring Databases: Why?



- Troubleshooting customer problems
  - Is there a routing problem?
- Analyzing equipment failures, planning system upgrades
  - When to add new routers, upgrade existing interfaces?

# Monitoring: What Can Go Wrong?

---

- Real world is dynamic, database may diverge from reality
- Configuration tables
  - Added new interface, configuration tables not updated
- Measurement tables
  - Added new interface, poller not informed → missing polls
  - Equipment failure → missing polls, anomalous values
- Solution: continuously audit the database using Data Auditor
  - Essential for data analysis to be meaningful!

# Outline

---

- Monitoring databases: what? how? why?
- Data audits: missing polls, inconsistencies
- Conditional constraints: tableaux generation

# What Is A Data Audit?

---

- Expert:
  - Specifies hypothesis assumed to be true (by analyst)
  - Akin to database integrity constraint
- System:
  - Identifies confidence of hypothesis in database
  - Identifies parsimonious description (tableau) of portions of database above or below specified confidence thresholds
- Expert:
  - Looks at tableaux, says “Aha, I know what’s wrong!” 😊

# Data Audit: Missing Polls

<u>tid</u>	<u>router</u>	<u>location</u>	<u>time</u>	<u>polls</u>
1	router1	New York	10:00	0
2	router2	New York	10:00	0
3	router3	Chicago	10:00	1
4	router4	Chicago	10:00	1
5	router1	New York	10:05	1
6	router2	New York	10:05	0
7	router3	Chicago	10:05	0
8	router4	Chicago	10:05	1
9	router1	New York	10:10	1
10	router2	New York	10:10	1
11	router3	Chicago	10:10	1
12	router4	Chicago	10:10	1
13	router1	New York	10:15	0
14	router2	New York	10:15	1
15	router3	Chicago	10:15	0
16	router4	Chicago	10:15	0

FOR EACH t in ROUTER\_POLLS  
ASSERT t.polls > 0

When/where is this assertion false  
(satisfied by at most 25% of rows)?

<u>router</u>	<u>location</u>	<u>time</u>	<u>conf</u>	<u>matches</u>

# Data Audit: Missing Polls

<u>tid</u>	<u>router</u>	<u>location</u>	<u>time</u>	<u>polls</u>
1	router1	New York	10:00	0
2	router2	New York	10:00	0
3	router3	Chicago	10:00	1
4	router4	Chicago	10:00	1
5	router1	New York	10:05	1
6	router2	New York	10:05	0
7	router3	Chicago	10:05	0
8	router4	Chicago	10:05	1
9	router1	New York	10:10	1
10	router2	New York	10:10	1
11	router3	Chicago	10:10	1
12	router4	Chicago	10:10	1
13	router1	New York	10:15	0
14	router2	New York	10:15	1
15	router3	Chicago	10:15	0
16	router4	Chicago	10:15	0

FOR EACH t in ROUTER\_POLLS  
ASSERT t.polls > 0

When/where is this assertion false  
(satisfied by at most 25% of rows)?

<u>router</u>	<u>location</u>	<u>time</u>	<u>conf</u>	<u>matches</u>
-	-	10:15	0.25	4

# Data Audit: Missing Polls

<u>tid</u>	<u>router</u>	<u>location</u>	<u>time</u>	<u>polls</u>
1	router1	New York	10:00	0
2	router2	New York	10:00	0
3	router3	Chicago	10:00	1
4	router4	Chicago	10:00	1
5	router1	New York	10:05	1
6	router2	New York	10:05	0
7	router3	Chicago	10:05	0
8	router4	Chicago	10:05	1
9	router1	New York	10:10	1
10	router2	New York	10:10	1
11	router3	Chicago	10:10	1
12	router4	Chicago	10:10	1
13	router1	New York	10:15	0
14	router2	New York	10:15	1
15	router3	Chicago	10:15	0
16	router4	Chicago	10:15	0

FOR EACH t in ROUTER\_POLLS  
ASSERT t.polls > 0

When/where is this assertion false  
(satisfied by at most 25% of rows)?

<u>router</u>	<u>location</u>	<u>time</u>	<u>conf</u>	<u>matches</u>
-	-	10:15	0.25	4
-	New York	10:00-10:05	0.25	4

# Data Audit: Inconsistencies

<u>tid</u>	<u>router</u>	<u>location</u>	<u>time</u>	<u>class</u>
1	router1	New York	10:00	0
2	router2	New York	10:00	1
3	router3	Chicago	10:00	2
4	router4	Chicago	10:00	2
5	router1	New York	10:05	0
6	router2	New York	10:05	1
7	router3	Chicago	10:05	2
8	router4	Chicago	10:05	2
9	router1	New York	10:10	0
10	router2	New York	10:10	1
11	router3	Chicago	10:10	1
12	router4	Chicago	10:10	1
13	router1	New York	10:15	0
14	router2	New York	10:15	1
15	router3	Chicago	10:15	2
16	router4	Chicago	10:15	2

FOR EACH t1, t2 in ROUTER\_CLASS  
WHERE t1.router = t2.router  
ASSERT t1.class = t2.class

When/where is this assertion (FD) false  
(satisfied by at most 50% of rows)?

<u>router</u>	<u>location</u>	<u>time</u>	<u>conf</u>	<u>matches</u>

# Data Audit: Inconsistencies

<u>tid</u>	<u>router</u>	<u>location</u>	<u>time</u>	<u>class</u>
1	router1	New York	10:00	0
2	router2	New York	10:00	1
3	router3	Chicago	10:00	2
4	router4	Chicago	10:00	2
5	router1	New York	10:05	0
6	router2	New York	10:05	1
7	router3	Chicago	10:05	2
8	router4	Chicago	10:05	2
9	router1	New York	10:10	0
10	router2	New York	10:10	1
11	router3	Chicago	10:10	1
12	router4	Chicago	10:10	1
13	router1	New York	10:15	0
14	router2	New York	10:15	1
15	router3	Chicago	10:15	2
16	router4	Chicago	10:15	2

FOR EACH t1, t2 in ROUTER\_CLASS  
WHERE t1.router = t2.router  
ASSERT t1.class = t2.class

When/where is this assertion (FD) false  
(satisfied by at most 50% of rows)?

<u>router</u>	<u>location</u>	<u>time</u>	<u>conf</u>	<u>matches</u>
-	Chicago	10:10-10:15	0.5	4

# Outline

---

- Monitoring databases: what? how? why?
- Data audits: missing polls, inconsistencies
- Conditional constraints: tableaux generation
  - Conditional functional dependencies [VLDB'08, SIGMOD'09]
  - Conditional sequential dependencies [VLDB'09]

# Problem Statement

---

- Given a table  $R$  and an FD, generate hold and fail tableaux
  - Philosophy: discover hidden semantics from data
- What makes for a good hold tableau?
  - High support, high confidence, parsimony
- Given a hold tableau, what makes for a good fail tableau?
  - High (residual) support, low (residual) confidence, parsimony

# Related Work

---

- Conditional functional dependencies
  - [BFG+07]: CFD definition, Armstrong axioms extension
  - [CFG+07]: violation detection, table repair to satisfy CFDs
  - [BFM07]: extended CFDs, inspiration for range tableaux
  - [CM08]: discovery of individual CFD patterns
  - [FGL+09]: faster discovery of individual CFD patterns
- Approximate FD discovery [HKPT99, KL03]
  - Useful to identify suitable embedded FDs

# Example: Config Table, FD

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

# Example: Config Table, CFD

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau [BFG+07]

<u>interf</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
-	routerC	-	-	-
-	routerB	09/11/09	-	0
-	-	10/11/09	-	-

# Example: Config Table, Fail Tableau

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau [BFG+07]

<u>interf</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
-	routerC	-	-	-
-	routerB	09/11/09	-	0
-	-	10/11/09	-	-

Fail Tableau [GKK+08]

<u>interf</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
-	-	11/11/09	-	-

# Metrics: Local Support, Confidence

- Given a CFD  $\Phi = (R: X \rightarrow Y, T_p)$  and table instance  $\text{adom}(R)$ 
  - $\text{Cover}(t_p) = \{t \mid (t \in \text{adom}(R)) \ \& \ (t[X] \text{ matches } t_p[X])\}$
- Local support of a pattern  $t_p$  in a tableau  $T_p$ 
  - $\text{LS}(t_p) = |\text{Cover}(t_p)|/|\text{adom}(R)|$
- Local confidence of a pattern  $t_p$  in a tableau  $T_p$ 
  - Let  $\text{Keepers}(t_p)$  denote records in  $\text{Cover}(t_p)$  after removing fewest records needed to eliminate all disagreements
  - $\text{LC}(t_p) = |\text{Keepers}(t_p)|/|\text{Cover}(t_p)|$

# Example: Local Metrics

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau

<u>interf</u>	<u>router</u>	<u>date</u>		<u>LS</u>	<u>LC</u>
-	routerC	-		0.35	6/7

# Example: Local Metrics

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau

<u>interf</u>	<u>router</u>	<u>date</u>	<u>LS</u>	<u>LC</u>
-	routerC	-	0.35	6/7
-	-	10/11/09	0.35	6/7

# Metrics: Global Support, Confidence

- Given a CFD  $\Phi = (R: X \rightarrow Y, T_p)$  and table instance  $\text{adom}(R)$
- Global support of a tableau  $T_p$ 
  - $\text{GS}(T_p) = |\mathbf{U} \text{ Cover}(t_p)|/|\text{adom}(R)|$
  - $\text{Max}(\text{LS}(t_p)) \leq \text{GS}(T_p) \leq \sum(\text{LS}(t_p))$
- Global confidence of a tableau  $T_p$ 
  - $\text{GC}(T_p) = |\mathbf{U} \text{ Keepers}(t_p)|/|\mathbf{U} \text{ Cover}(t_p)|$
  - Possible that  $\text{GC}(T_p) \leq \min(\text{LC}(t_p))$
  - Possible that  $\text{GC}(T_p) \geq \max(\text{LC}(t_p))$

# Example: Global Metrics

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau: GS = 0.5, GC = 0.8

<u>interf</u>	<u>router</u>	<u>date</u>	<u>LS</u>	<u>LC</u>
-	routerC	-	0.35	6/7
-	-	10/11/09	0.35	6/7

# Tableau Generation Problem 1

- Given a CFD  $\Phi = (R: X \rightarrow Y, T_p)$ , table instance  $\text{adom}(R)$  is
  - $(s,c)_{\text{gg}}$ -satisfied by  $\Phi$  iff  $\text{GS}(T_p) \geq s$  and  $\text{GC}(T_p) \geq c$
- Tableau generation problem with GS and GC
  - Given FD  $R: X \rightarrow Y$ ,  $\text{adom}(R)$ , and  $(s,c)$  find  $T_p$  of smallest size such that  $\text{adom}(R)$  is  $(s,c)_{\text{gg}}$ -satisfied by  $(R: X \rightarrow Y, T_p)$
- Complexity of problem
  - NP-complete
  - Provably hard to approximate within  $|\text{adom}(R)|^{1/2 - \epsilon}$ ,  $\epsilon > 0$

# Tableau Generation Problem 2

- Given a CFD  $\Phi = (R: X \rightarrow Y, T_p)$ , table instance  $\text{adom}(R)$  is
  - $(s,c)_{gl}$ -satisfied by  $\Phi$  iff  $GS(T_p) \geq s$  and for all  $t_p \in T_p$ ,  $LC(t_p) \geq c$
- Tableau generation problem with GS and LC
  - Given FD  $R: X \rightarrow Y$ ,  $\text{adom}(R)$ , and  $(s,c)$  find  $T_p$  of smallest size such that  $\text{adom}(R)$  is  $(s,c)_{gl}$ -satisfied by  $(R: X \rightarrow Y, T_p)$
- Complexity of problem
  - NP-complete: reduction from vertex cover in tripartite graphs
  - Provably hard to approximate to within  $34/33$

# Problem 2: Greedy Approximation

- Problem with GS and LC can be reduced to Partial Set Cover
  - Partial Set Cover admits a greedy approximation algorithm
- Reduction
  - Generate all candidate patterns (data cube) from  $\text{adom}(R)$
  - Iteratively choose pattern with highest marginal local support, satisfying  $\text{LC}(t) \geq c$
  - Stop when (hold) tableau's  $\text{GS} \geq s$
- Approximation guarantee, complexity of greedy approximation
  - $|T|/|T^*| \leq 1 + \ln(s \cdot |\text{adom}(R)|)$ , compared to optimal  $T^*$
  - Complexity is  $O(2^K \cdot |\text{adom}(R)|)$ ,  $K = \text{antecedents}(\text{FD})$

# Issues with Greedy Algorithm

---

- Very high initial cost
  - Generate **all** candidate patterns (data cube) from  $\text{adom}(R)$
  - Example: 48 candidate patterns generated, most useless
- Very high incremental cost
  - Iteratively maintain **all** (even unused) marginal local supports
  - Example: 3 iterations
- Good news
  - Initial and incremental costs can be substantially reduced

# Problem 2: On-demand Algorithm

---

- Key observation: generate candidate patterns only as needed
- Algorithm
  - Start with the “all wildcards” candidate pattern in frontier
  - Iteratively visit frontier patterns in decreasing MLS order
  - If candidate pattern meets LC threshold, include in tableau else consider adding its “children” patterns to frontier
  - Important: generate a pattern only when all “parents” visited
  - Stop when (hold) tableau’s  $GS \geq s$
- Result: correspondence with (off-demand) greedy algorithm
  - Same patterns chosen for tableau in same order

# Example: On-demand Algorithm

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau: GS = 0.75, LC = 0.8 , CGS = 0.0

<u>interf</u>	<u>router</u>	<u>date</u>	<u>MLS</u>	<u>LC</u>
-	routerB	-	0.35	5/7
-	routerC	-	0.35	6/7
-	-	10/11/09	0.35	6/7
-	-	11/11/09	0.25	0.4
-	routerD	-	0.3	0.5
<del>-</del>	<del>-</del>	<del>09/11/09</del>	<del>0.4</del>	<del>6/8</del>

# Example: On-demand Algorithm

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau: GS = 0.75, LC = 0.8 , CGS = 0.0

<u>interf</u>	<u>router</u>	<u>date</u>	<u>MLS</u>	<u>LC</u>
-	routerB	-	0.35	5/7
-	routerC	-	0.35	6/7
-	-	10/11/09	0.35	6/7
-	-	11/11/09	0.25	0.4
-	routerD	-	0.3	0.5
-	-	09/11/09	0.4	6/8
-	routerB	09/11/09	0.25	0.8

# Example: On-demand Algorithm

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau: GS = 0.75, LC = 0.8 , CGS = 0.35

<u>interf</u>	<u>router</u>	<u>date</u>	<u>MLS</u>	<u>LC</u>
-	routerB	-	0.35	5/7
-	routerC	-	0.35	6/7
-	-	10/11/09	0.15	6/7
-	-	11/11/09	0.25	0.4
-	routerD	-	0.3	0.5
-	-	09/11/09	0.4	6/8
-	routerB	09/11/09	0.25	0.8

# Example: On-demand Algorithm

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau: GS = 0.75, LC = 0.8 , CGS = 0.35

<u>interf</u>	<u>router</u>	<u>date</u>	<u>MLS</u>	<u>LC</u>
-	routerB	-	0.35	5/7
-	routerC	-	0.35	6/7
-	-	10/11/09	0.15	6/7
-	-	11/11/09	0.25	0.4
-	routerD	-	0.3	0.5
-	-	09/11/09	0.4	6/8
-	routerB	09/11/09	0.25	0.8
-	routerB	11/11/09	0.1	0.5
-	routerD	11/11/09	0.15	1/3

# Example: On-demand Algorithm

<u>tid</u>	<u>interface</u>	<u>router</u>	<u>date</u>	<u>ip</u>	<u>type</u>
1	interface1	routerB	09/11/09	10.30.15.10	0
2	interface1	routerB	09/11/09	10.30.15.10	0
3	interface1	routerB	09/11/09	10.30.15.10	5
4	interface2	routerB	09/11/09	10.30.15.25	0
5	interface2	routerB	09/11/09	10.30.15.25	0
6	interface3	routerB	11/11/09	10.30.15.30	4
7	interface3	routerB	11/11/09	10.30.15.40	4
8	interface4	routerC	10/11/09	10.30.15.255	5
9	interface4	routerC	10/11/09	10.30.15.255	5
10	interface5	routerC	10/11/09	10.30.15.250	0
11	interface5	routerC	10/11/09	10.30.15.250	0
12	interface6	routerC	09/11/09	10.30.15.200	5
13	interface6	routerC	09/11/09	10.30.15.200	5
14	interface6	routerC	09/11/09	10.30.15.255	5
15	interface7	routerD	10/11/09	10.30.15.19	0
16	interface8	routerD	10/11/09	10.30.15.29	0
17	interface8	routerD	10/11/09	10.30.15.225	0
18	interface9	routerD	11/11/09	10.30.15.225	8
19	interface9	routerD	11/11/09	10.30.15.225	0
20	interface9	routerD	11/11/09	10.30.15.220	0

FD: [interface, router, date] → [ip, type]

Confidence = 0.70

Hold Tableau: GS = 0.75, LC = 0.8 , CGS = 0.75

<u>interf</u>	<u>router</u>	<u>date</u>	<u>MLS</u>	<u>LC</u>
-	routerB	-	0.35	5/7
-	routerC	-	0.35	6/7
-	-	10/11/09	0.15	6/7
-	-	11/11/09	0.25	0.4
-	routerD	-	0.3	0.5
-	-	09/11/09	0.4	6/8
-	routerB	09/11/09	0.25	0.8
-	routerB	11/11/09	0.1	0.5
-	routerD	11/11/09	0.15	1/3

# Generating Range Tableaux

---

- Difference between (ordinary) tableau and range tableau
  - Permit ranges  $[a_l, a_r]$  in ordered attributes of patterns
  - Much larger number of candidate patterns
  - Allows for substantially more parsimonious tableaux
- Can reuse on-demand algorithm with suitable changes
  - Children of  $[a_l, a_r]$  are  $[a_l, a_{r-1}]$  and  $[a_{l+1}, a_r]$

# Example: Range Tableau

<u>interface</u>	<u>router</u>	<u>date</u>		<u>ip</u>	<u>type</u>
-	routerC	-		-	-
-	routerB	[01/11/09, 09/11/09]		-	0
-	routerB	[10/11/09, 11/11/09]		-	5
-	-	[05/11/09, 08/11/09]		-	-

# Experiments: Real Data Sets

- 300K sales records from online retailer
  - Sales(tid, itemid, name, type, price, tax, country, city)

FD1	type, name, country → price, tax, itemid
-----	--

- 30-day excerpt of network router configuration table
  - Config(date, router, interface, interface\_type, ip\_address)

FD2	router, interface → ip_address
FD3	router, interface, interface_type → ip_address
FD4	router, interface, date → ip_address

# Experiments: Hold Tableau Sizes

- Summary of hold tableau for FD1,  $lc = 0.88$

support threshold		size	optimal size	global confidence
0.3		1	1	0.908
0.4		2	2	0.916
0.5		2	2	0.916
0.6		2	2	0.916
0.7		3	3	0.922
0.8		41	41	0.924
0.9		1690	1689	0.927

# Experiments: Performance

- Comparison of running times, number of patterns considered

<u>GS</u>	<u>Time</u> <u>Off-demand</u>	<u>Time</u> <u>On-demand</u>		<u>Patterns</u> <u>Off-demand</u>	<u>Patterns</u> <u>On-demand</u>
0.5	11.5s	5.2s		610	90
0.7	11.8s	5.4s		610	92
0.8	12.0s	5.9s		610	150
0.85	12.2s	6.0s		610	155
0.9	12.5s	6.5s		610	190

# Experiments: Range Tableau Size

- Reducing tableau size for FD4 with attribute ranges

<u>Support Threshold</u>		<u>Tableau Size</u>	<u>Tableau size with ranges</u>
0.5		23	1
0.6		76	1
0.7		328	2
0.8		2634	4
0.9		N/A	320

# Experiments: Summary

---

- Greedy algorithms return near-optimal tableaux
  - Far smaller than upper bound of approximation guarantee
- On-demand algorithm much faster than off-demand algorithm
  - Difference increases with number of candidate patterns
- Range tableaux much smaller than standard tableaux
  - If embedded FD holds for a range of antecedent values

# Summary

---

- Data quality a serious issue in today's monitoring databases
  - Inconsistencies, missing and delayed polls, anomalies
  - Data audits help by identifying potential data quality issues
- Key contributions: generating good tableaux
  - Optimal tableau definition: support, confidence, parsimony
  - Tableaux discovery for CFDs [VLDB'08, SIGMOD'09]
  - Tableaux discovery for CSDs [VLDB'09]
  - Data Auditor in current use in our monitoring databases