

Privacy of Numeric Queries Via Simple Value Perturbation

The Laplace Mechanism

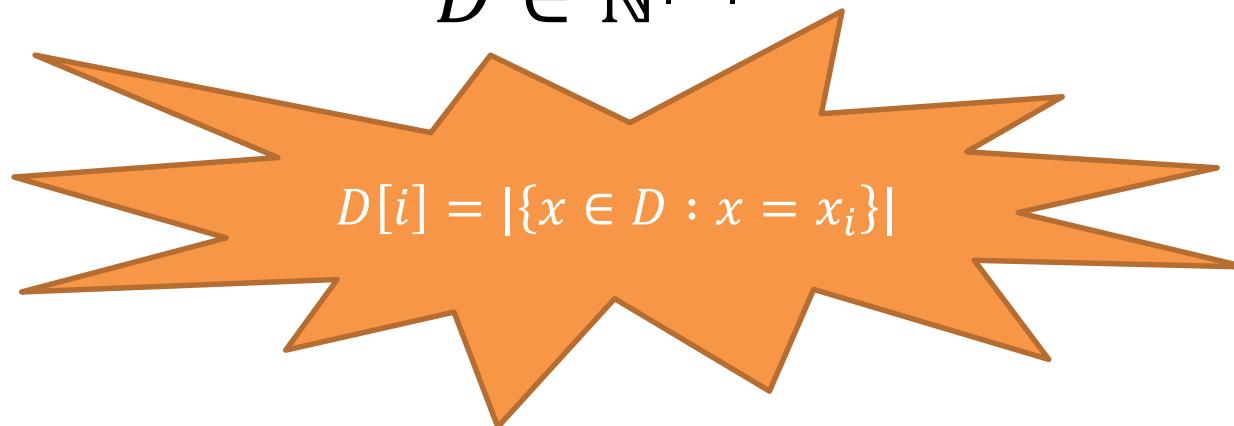
Slides are extracted from material by Aaron Roth

Differential Privacy

A Basic Model

- Let X represent an abstract data universe and D be a multi-set of elements from X .
 - i.e. D can contain multiple copies of an element $x \in X$.
- Convenient to represent D as a *histogram*:

$$D \in \mathbb{N}^{|X|}$$



Differential Privacy

A Basic Model

- i.e for a database of heights
 - $D = \{5'2, 6'1, 5'8, 5'8, 6'0\} \subset [4 - 8]$
 - $D = (\dots, 1, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 1, 1, 0, \dots) \in \mathbb{R}^{48}$ 

Differential Privacy

A Basic Model

- The *size* of a database n :
 - As a set: $n = |D|$.
 - As a histogram: $n = ||D||_1 = \sum_{i=1}^{|X|} |D[i]|$

Definition: ℓ_1 (Manhattan) Distance.

For $\hat{v} \in \mathbb{R}^d$, $||\hat{v}||_1 = \sum_{i=1}^d |\hat{v}_i|$.

Differential Privacy

A Basic Model

- The *distance* between two databases:
 - As a set: $|D \Delta D'|$.
 - As a histogram: $\|D - D'\|_1$

Differential Privacy

A Basic Model

- i.e for a database of heights
 - $D = \{5'2, 6'1, 5'8, 5'8, 6'0\} \subset [4 - 8]$
 - $D = (\dots, 1, 0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 1, 0, \dots) \in \mathbb{R}^{48}$


5'2 5'8 6'0 6'1
 - $D' = (\dots, 2, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, \dots) \in \mathbb{R}^{48}$

$$\|D\|_1 = |1| + |2| + |1| + |1| = 5$$

$$\|D'\|_1 = |2| + |1| + |1| + |1| + |1| = 6$$

$$\|D - D'\|_1 = |-1| + |-1| + |1| = 3$$

Basic Lower Bound: Blatant Non-Privacy

- How much noise is necessary to guarantee privacy?
- A simple model.
 - For simplicity, $D \in \{0,1\}^{|X|}$ (i.e. no repeated elts)
 - A query is a bit vector $Q \in \{0,1\}^{|X|}$
 - $Q(D) = \langle Q, D \rangle = \sum_{i:Q[i]=1} D[i]$
 - A “subset sum query”
 - For $S \subseteq [n]$ write Q_S for the vector:
$$Q_S[i] = \begin{cases} 1, & i \in S \\ 0, & i \notin S \end{cases}$$

Differential Privacy

A Basic Model

Definition: A randomized algorithm with domain $\mathbb{N}^{|X|}$ and range R

$$M: \mathbb{N}^{|X|} \rightarrow R$$

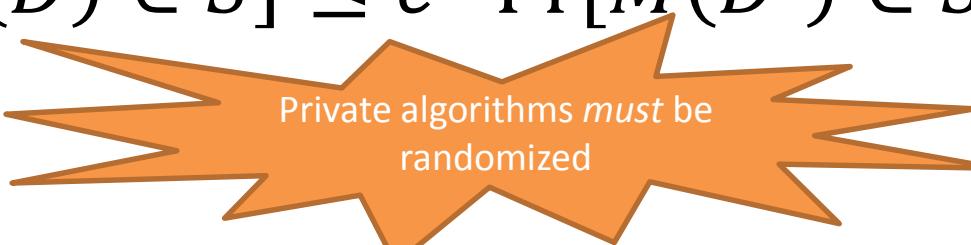
is (ϵ, δ) -differentially private if:

- 1) For all pairs of databases $D, D' \in \mathbb{N}^{|X|}$ such that $\|D - D'\|_1 \leq 1$ and,
- 2) For all events $S \subseteq R$:

$$\Pr[M(D) \in S] \leq e^\epsilon \Pr[M(D') \in S] + \delta.$$



Differing in 1 person's data



Private algorithms must be randomized

Resilience to Post Processing

Proposition: Let $M: \mathbb{N}^{|X|} \rightarrow R$ be (ϵ, δ) -differentially private and let $f: R \rightarrow R'$ be an arbitrary function. Then:

$$f \circ M: \mathbb{N}^{|X|} \rightarrow R'$$

is (ϵ, δ) -differentially private.



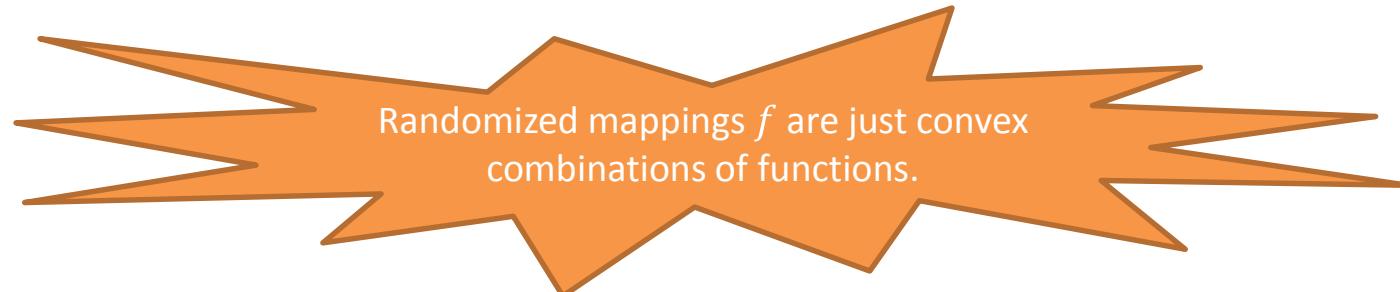
Resilience to Post Processing

Proof:

- 1) Consider any pair of databases $D, D' \in \mathbb{N}^{|X|}$ with $\|D - D'\|_1 \leq 1$.
- 2) Consider any event $S \subseteq R'$.
- 3) Let $T \subseteq R$ be defined as $T = \{r \in R : f(r) \in S\}$.

Now:

$$\begin{aligned}\Pr[f(M(D)) \in S] &= \Pr[M(D) \in T] \\ &\leq e^\epsilon \Pr[M(D') \in T] + \delta \\ &= e^\epsilon \Pr[f(M(D)) \in S] + \delta\end{aligned}$$



Resilience to Post Processing

Take away message:

- 1) f as the adversaries analysis: can incorporate arbitrary auxiliary information the adversary may have. Privacy guarantee holds no matter what he does.
- 2) f as our algorithm: If we access the database in a differentially private way, we don't have to worry about how our algorithm post-processes the result. *We only have to worry about the data access steps.*

Answering Numeric Queries

- Suppose we have some numeric *question* about the private database that we want to know the answer to:

$$Q: \mathbb{N}^{|X|} \rightarrow \mathbb{R}^k. \quad Q(D) = ?$$

- How do we do it privately?
 - How much noise do we have to add?
 - What are the relevant properties of Q ?

Answering Numeric Queries

Definition: The ℓ_1 -sensitivity of a query

$Q: \mathbb{N}^{|X|} \rightarrow \mathbb{R}^k$ is:

$$GS(Q) = \max_{D, D': \left\| D - D' \right\|_1 \leq 1} \left\| Q(D) - Q(D') \right\|_1$$

i.e. how much can 1 person affect the value of the query?

“How many people in this room have brown eyes”: Sensitivity 1

“How many have brown eyes, how many have blue eyes, how many have green eyes, and how many have red eyes”: Sensitivity 1

“How many have brown eyes and how many are taller than 6”: Sensitivity 2

Answering Numeric Queries

The Laplace Distribution:

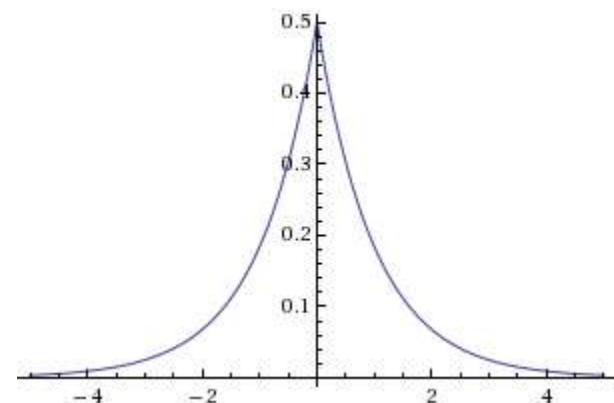
$\text{Lap}(b)$ is the probability distribution with p.d.f.:

$$p(x | b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$$

i.e. a symmetric exponential distribution

$$Y \sim \text{Lap}(b), \quad E[|Y|] = b$$

$$\Pr[|Y| \geq t \cdot b] = e^{-t}$$



Answering Numeric Queries: The Laplace Mechanism

$\text{Laplace}(D, Q: \mathbb{N}^{|X|} \rightarrow \mathbb{R}^k, \epsilon)$:

1. Let $\Delta = GS(Q)$.
2. For $i = 1$ to k : Let $Y_i \sim \text{Lap}(\frac{\Delta}{\epsilon})$.
3. Output $Q(D) + (Y_1, \dots, Y_k)$

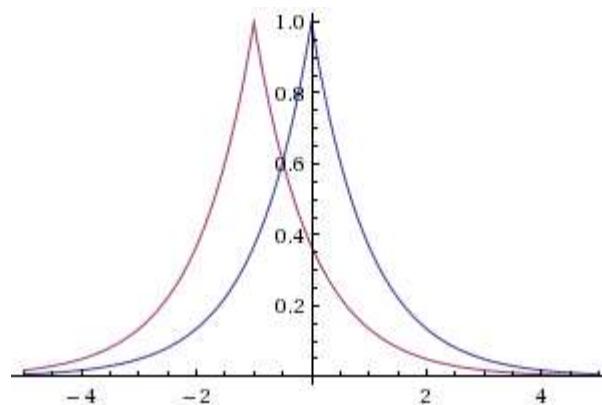
Independently perturb each coordinate of the output with Laplace noise scaled to the sensitivity of the function.

Idea: This should be enough noise to hide the contribution of any single individual, no matter what the database was.

Answering Numeric Queries: The Laplace Mechanism

$\text{Laplace}(D, Q: \mathbb{N}^{|X|} \rightarrow \mathbb{R}^k, \epsilon)$:

1. Let $\Delta = GS(Q)$.
2. For $i = 1$ to k : Let $Y_i \sim \text{Lap}(\frac{\Delta}{\epsilon})$.
3. Output $Q(D) + (Y_1, \dots, Y_k)$



To Ponder

- Where is there room for improvement?
 - The Laplace mechanism adds *independent* noise to every coordinate...
 - What happens if the user asks (essentially) the same question in every coordinate?
 - Read [Dinur,Nissim03]: a computationally efficient attack that gives blatant non-privacy for a mechanism that adds noise bounded by $o(\sqrt{n})$.

The Algorithmic Foundations of Data Privacy

Instructor: Aaron Roth

Material obtained from <https://www.cis.upenn.edu/~aaroth/courses/privacyF11.html>

Free access textbook: <https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf>

Today

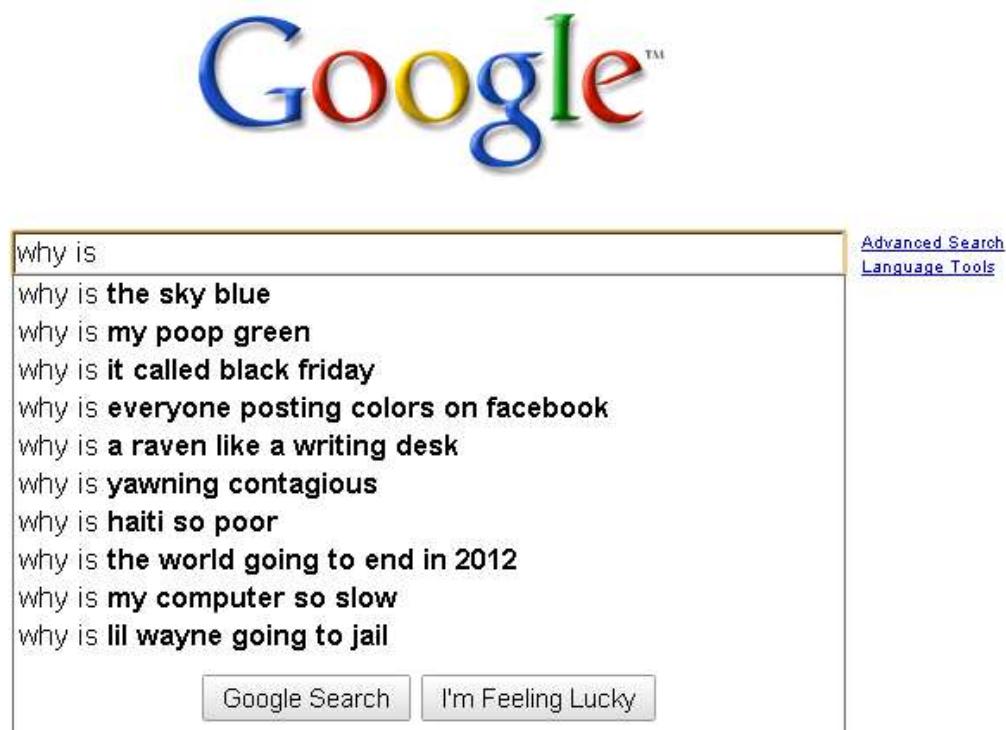
- Some motivation
- The definition of differential privacy
- An overview of topics we will cover
- If there is time: A lower bound.

Modern Algorithm Design

- Computation is not the only constraint
- Dealing with large datasets
 - Data *belongs* to other people
 - Must protect their privacy
 - Must convince them to report it truthfully

Modern Algorithm Design

- Use search logs to recommend query completions



Modern Algorithm Design

- Find closely connected components in a social network

 **Suggestions**
Add people you know as friends and become a fan of public profiles you like.

 Joe Locaccino Add as friend	 Marie F. Add as friend	 Severin Hacker Add as friend
 Gregory Sorkin Add as friend	 Jim McCann Add as friend	 Jonathan Derryberry Add as friend
 Andreas Krause Add as friend	 Kevin Bierhoff Add as friend	 Danny Sleator Add as friend
 Ajit Singh Add as friend	 Alan Michael Friese Add as friend	 Polo Chau Add as friend
 Swapnil Patil Add as friend	 Michael Verde Add as friend	 Frank Pfenning Add as friend
 Johannes Schmieder Add as friend	 Benoit Hudson Add as friend	 Stephen Magill Add as friend
 Nikhil Bansal Add as friend	 Dan Licata Add as friend	 Colin McMillen Add as friend
 Reza Zadeh (Reza Bosagh Zadeh) Add as friend	 David Brumley Add as friend	 Liz Crawford Add as friend
 Himanshu Jain Add as friend	 Rob Reeder Add as friend	

Modern Algorithm Design

- Decide which ads to show based on user data and other users previous searches.

The screenshot shows a Google search results page. The search query 'FOCS 07' is entered in the search bar. The results are categorized under 'Web'. The first result is a news article from 'Computational Complexity' about the FOCS Day 1 and Business Meeting. The second result is another news article from the same source about the STOC and FOCS meeting. Both results include links to the full articles, cached versions, and similar pages. To the right of the search results, there is a 'Sponsored Links' sidebar with an advertisement for '2007 Fox' from AutoTrader.com. A callout box highlights the 'Pittsburgh Soda' entry in the sidebar, which offers local soda search services for Pittsburgh, PA.

Google™ Web Images Video News Maps Desktop more »

FOCS 07

Search Advanced Search Preferences

Web Results 1 - 10 of about 136,000 for FOCS 07. (0.09 seconds)

[Computational Complexity: FOCS Day 1 and Business Meeting](#) - 2:39pm
FOCS 07 will be in Providence, RI. Program chair is Alistair Sinclair. Location is the new Renaissance Hotel, currently under construction. ...
[weblog.fortnow.com/2006/10/focs-day-1-and-business-meeting.html](#) - 34k -
[Cached](#) - [Similar pages](#) - [Note this](#)

[Computational Complexity: STOC and FOCS](#)
And in that great circle of theory life, the FOCS '07 Call for Papers is out.
Submission deadline is April 20 and FOCS will be held October 21-23 in ...
[weblog.fortnow.com/2007/02/stoc-and-focs.html](#) - 28k -
[Cached](#) - [Similar pages](#) - [Note this](#)
[More results from [weblog.fortnow.com](#)]

Sponsored Links

[2007 Fox](#)
Make Volkswagen Shopping Easier
Compare Your Favorites Side By Side
[www.AutoTrader.com](#)

[Pittsburgh Soda](#)
Find soda here!
We offer local search in your city
[Pittsburgh.Local.com](#)
Pittsburgh, PA

What is Privacy?



What Isn't Privacy?

- Privacy isn't restricting questions to large populations.
 - “What is the average salary of Penn faculty?”
 - “What is the average salary of Penn faculty not named Aaron Roth?”

What Isn't Privacy?

- Privacy isn't restricting to “ordinary” facts.
 - Statistics on Alice’s bread buying habits: For 20 years she regularly buys bread, and then stops.
 - Type 2 diabetes?

What Isn't Privacy?

- Privacy isn't “Anonymization”
 - Anonymization is hard.
 - Problem: Auxiliary Information and Linkage Attacks!
 - Case Study: NetFlix Prize Dataset
 - Linked with IMDB database to re-identify users [Narayanan, Shmatikov]
 - 2nd Netflix prize cancelled
 - Can't know what the adversary knows, or might know in the *future*.

What Isn't Privacy?

- Privacy isn't “Anonymization”
 - Anonymization isn't enough
 - Collection of medical records from a specific urgent care center and date might correspond to only a small collection of medical conditions.
 - Knowledge (from a neighbor?) that Alice went to that urgent care center doesn't identify her record, but implies she has one of a small number of conditions.

What is Privacy?

- Freedom from harm.

Privacy Definition, Attempt 1:

An analysis of a dataset D is private if the data analyst knows no more about Alice after the analysis than he knew about Alice before the analysis.

What is Privacy

- Problem: Impossible to achieve with auxiliary information.
 - Suppose an insurance company knows that Alice is a smoker.
 - An analysis that reveals that smoking and lung cancer are correlated might cause them to raise her rates!
- Was her privacy violated?
 - This is a problem *even if Alice was not in the database!*
 - This is exactly the sort of information we want to be able to learn...

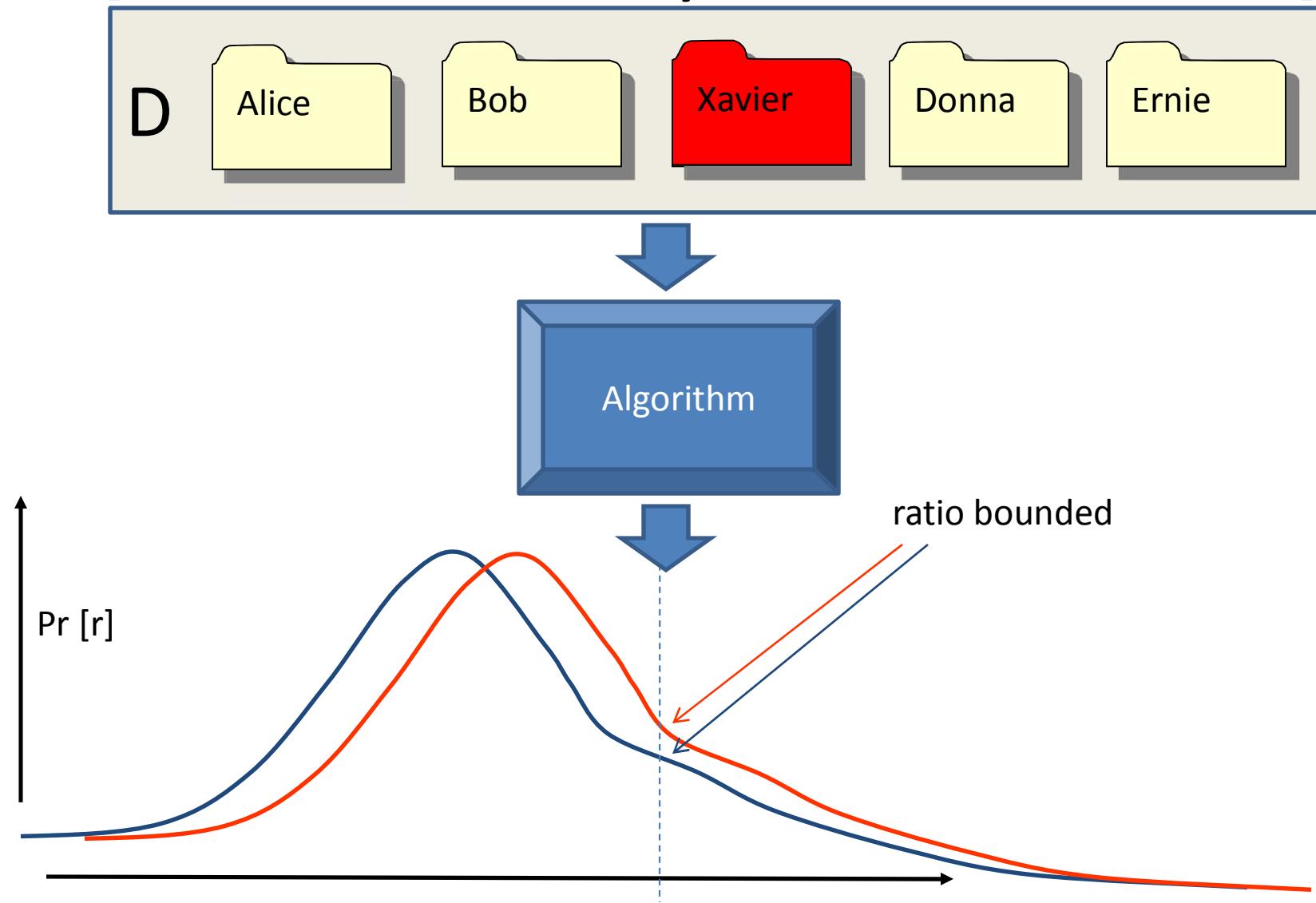
What is Privacy?

Privacy Definition, Attempt 2:

*An analysis of a dataset D is private if the data analyst knows **almost** no more about Alice after the analysis than he **would have known had he conducted the same analysis on an identical database with Alice's data removed.***

Differential Privacy

[Dwork-McSherry-Nissim-Smith 06]



Differential Privacy

X : The data *universe*.

$D \subset X$: The dataset (one element per person)

Definition: Two datasets $D, D' \subset X$ are *neighbors* if they differ in the data of a single individual. i.e. $|D \Delta D'| \leq 1$.

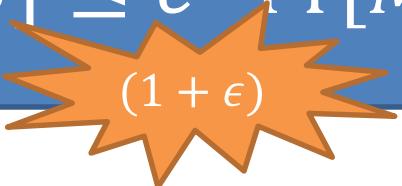
Differential Privacy

X : The data *universe*.

$D \subset X$: The dataset (one element per person)

Definition: A mechanism $M: 2^X \rightarrow R$ is (ϵ, δ) -differentially private if for all pairs of neighboring databases $D, D' \subset X$, and for all events $S \subseteq R$:

$$\Pr[M(D) \in S] \leq e^\epsilon \Pr[M(D') \in S] + \delta$$



$(1 + \epsilon)$

Definition: A mechanism $M: 2^X \rightarrow R$ is (ϵ, δ) -differentially private if for all pairs of neighboring databases $D, D' \subset X$, and for all events $S \subseteq R$:

$$\Pr[M(D) \in S] \leq e^\epsilon \Pr[M(D') \in S] + \delta$$

- Think of δ as exponentially small (or even 0)
- Think of ϵ as a small constant.
 - If $M: 2^X \rightarrow R$ is $(\epsilon, 0)$ -DP, and $|D \Delta D'| = k$, then:
$$\Pr[M(D) \in S] \leq e^{\epsilon k} \Pr[M(D') \in S]$$
 - So nothing useful is possible for $\epsilon = o(\frac{1}{n})$

Why is Differential Privacy “Privacy”?

- It should guarantee “freedom from harm”
- A useful fact – resilience to post-processing:
 - For any $f: R \rightarrow R'$, and any (ϵ, δ) -differentially private $M: 2^X \rightarrow R$, $f \circ M: 2^X \rightarrow R'$ is also (ϵ, δ) -differentially private.
- What if f maps mechanism output to events you care about?
 - Differential privacy: “Except for rare events that occur with probability $\leq \delta$, your future utility will decrease by at most a $(1 - \epsilon)$ factor by participating in the database.”

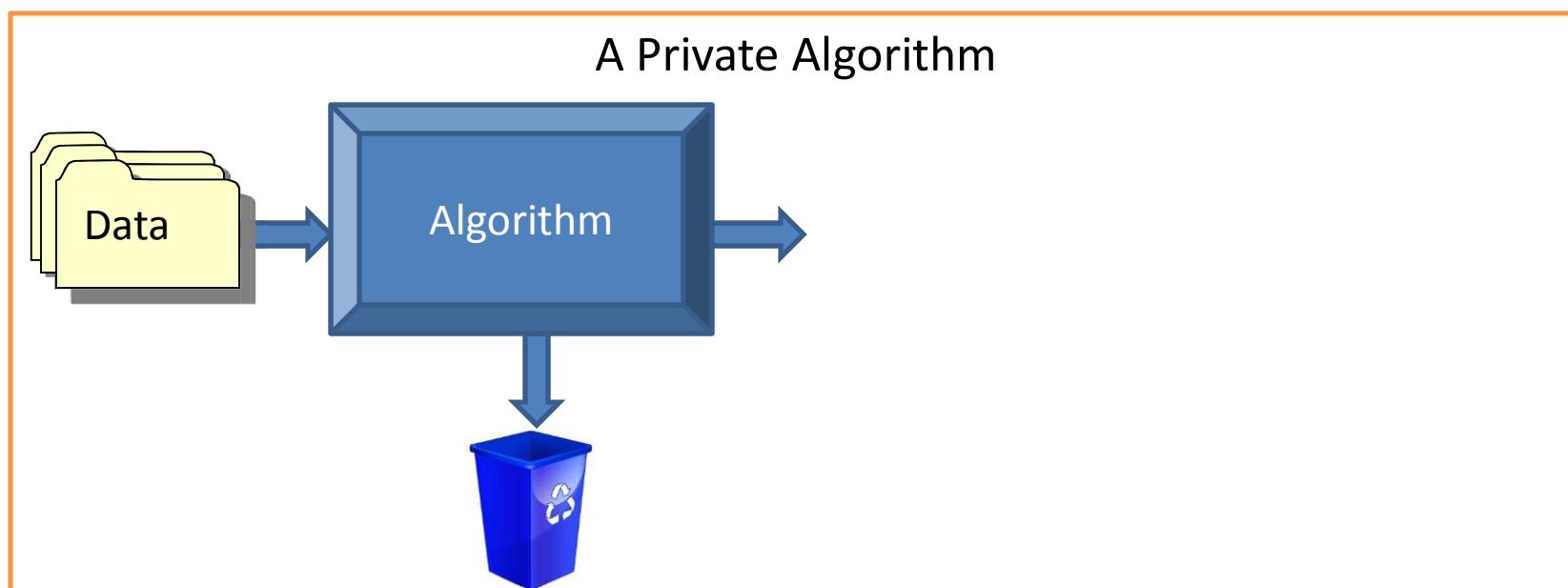
Why is Differential Privacy “Privacy”?

- f incorporates any auxiliary information an analyst may have about the database now *or in the future*.
- The guarantee is just as strong *even if the analyst knows the entire database except for your value*.
 - A worst case model: no longer any need to reason about what the analyst knows.

So now we have a definition.

Course Roadmap

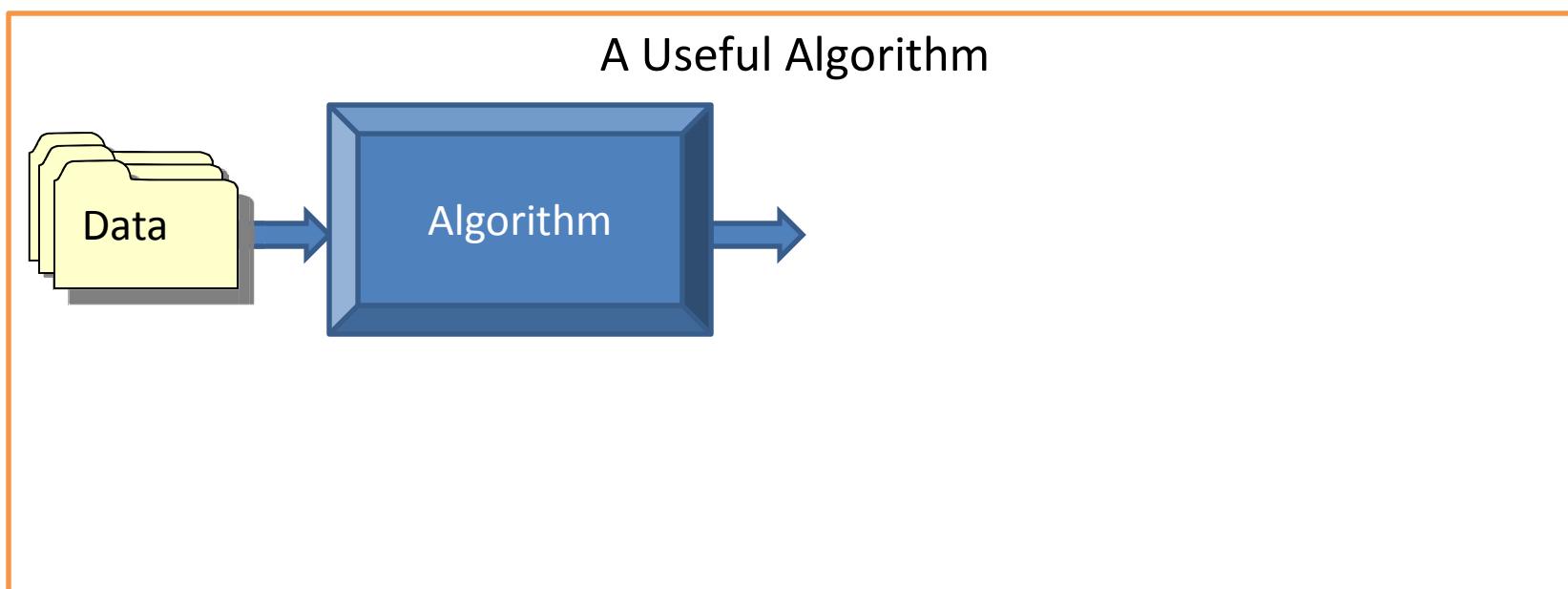
- What are the big questions?
 - How do we trade off privacy and utility?



So now we have a definition.

Course Roadmap

- What are the big questions?
 - How do we trade off privacy and utility?



So now we have a definition.

Course Roadmap

- How can we build useful, differentially private algorithms?
 - Out of basic building blocks, glued together by composition theorems.

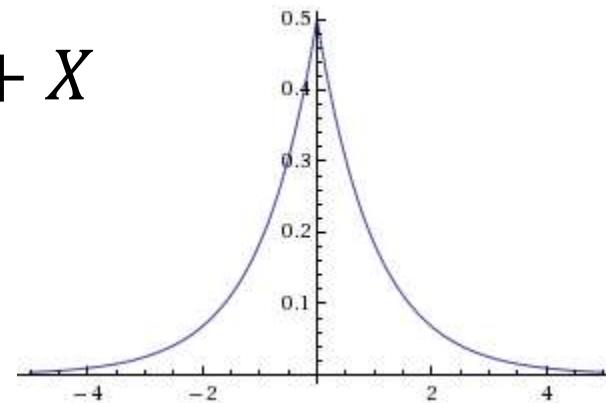
So now we have a definition.

Course Roadmap

- Basic Building Blocks
 - Answering numeric queries through perturbation

$$M_f(D) = f(D) + X$$

$$X \sim Lap\left(\frac{1}{\epsilon}\right)$$



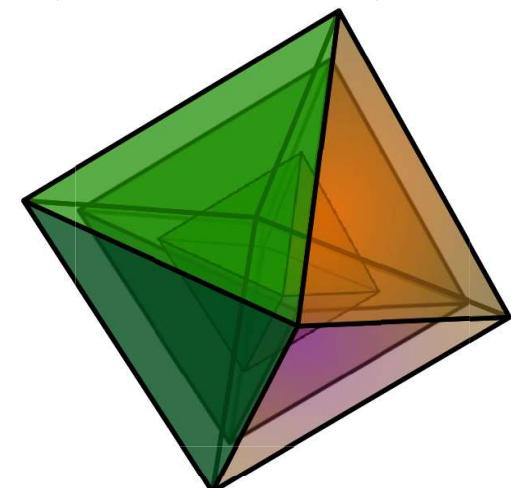
So now we have a definition.

Course Roadmap

- Basic Building Blocks
 - Answering non-numeric queries by sampling from a private distribution

$$M_q(D) : 2^X \rightarrow R$$

Output $r \in R$ with probability $\sim \exp(-\epsilon q(r, D))$



So now we have a definition.

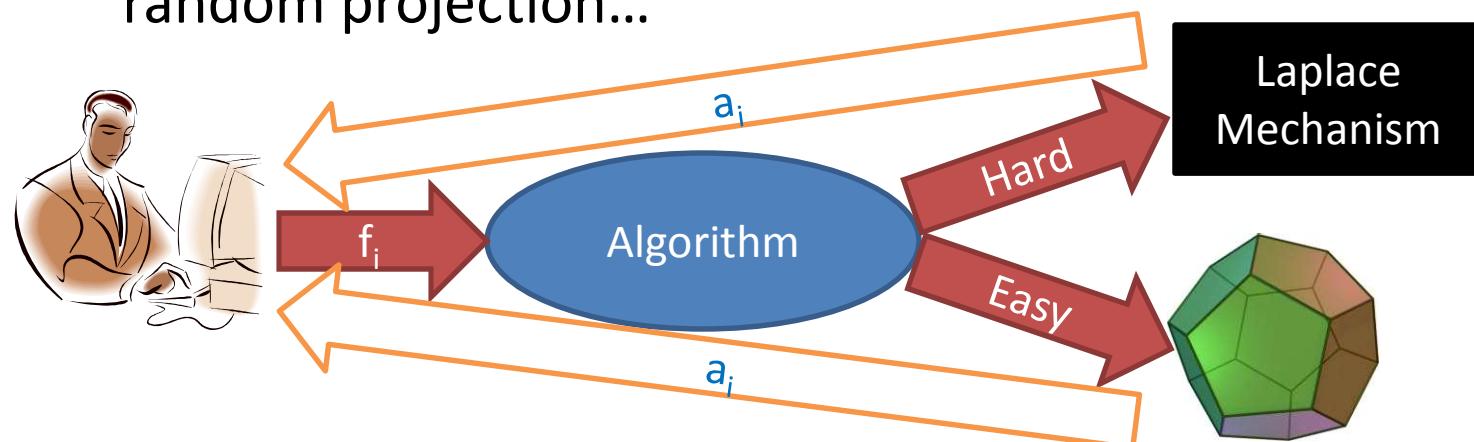
Course Roadmap

- Combining building blocks into algorithms
 - What are the privacy guarantees for an algorithm M composed of k subroutines A_1, \dots, A_k that are each (ϵ, δ) -differentially private?
 - $(k\epsilon, k\delta)$ -differentially private
 - $Also \approx (\sqrt{k \log \frac{1}{\delta'}}, \epsilon, k\delta + \delta')$ -differentially private
 - Can *trade* lots of ϵ for a little more δ .

So now we have a definition.

Course Roadmap

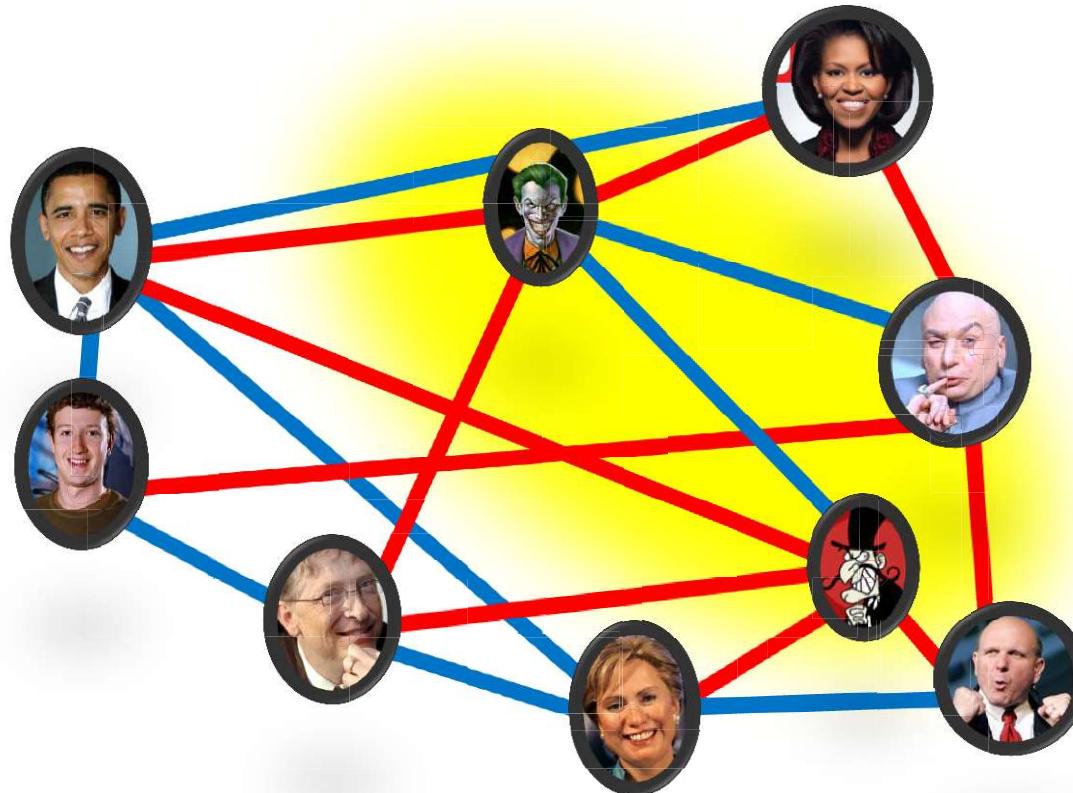
- What can we build?
 - Algorithms for accurately answering *exponentially* many numeric queries in the database size!
 - Leveraging machine learning theory, compression, random projection...



So now we have a definition.

Course Roadmap

- What can we build?
 - Algorithms for combinatorial optimization



So now we have a definition.

Course Roadmap

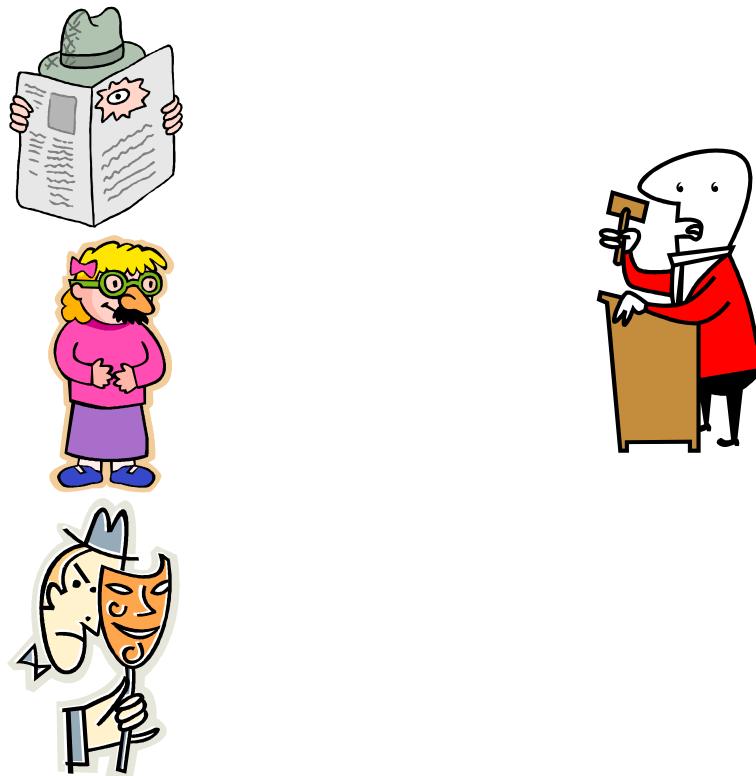
- What can we build?
 - Streaming Algorithms
 - That are private even if a hacker is able to look at the internal state of the algorithm.



So now we have a definition.

Course Roadmap

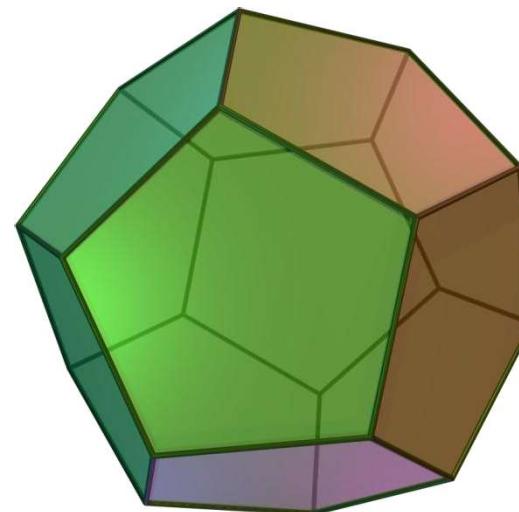
- What can we build?
 - Auctions and truthful mechanisms for privacy-aware economic agents



So now we have a definition.

Course Roadmap

- What *can't* we build?
 - Lower bounds from linear programming
 - Answering queries *too* accurately lets an adversary reconstruct the database



So now we have a definition.

Course Roadmap

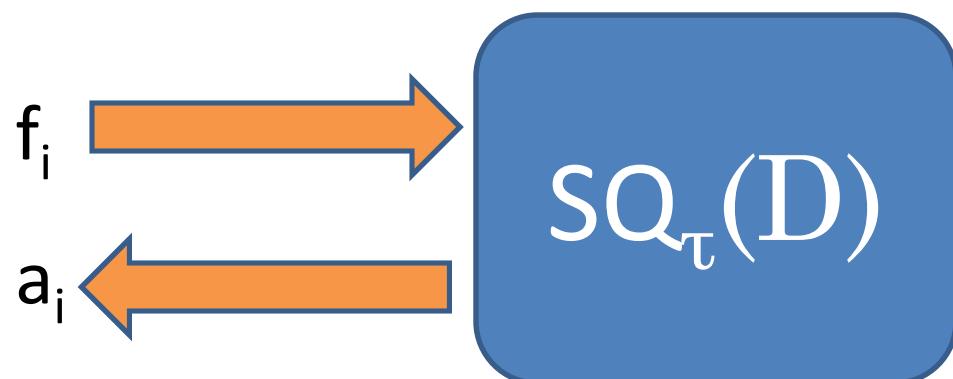
- What *can't* we build?
 - Lower bounds from packing arguments
 - The existence of good *error correcting codes* give lower bounds in differential privacy



So now we have a definition.

Course Roadmap

- What *can't* we build?
 - Lower bounds from learning theory
 - Efficient query release algorithms in Kearns' *statistical query* model would lead to too-good-to-be-true learning algorithms.



To Muse On:

- Think about why differential privacy protects against blatant non-privacy
- Read [Narayanan,Shmatikov06]: How to de-anonymize the Netflix data set.

A Short Tutorial on Differential Privacy

Borja Balle

Amazon Research Cambridge

The Alan Turing Institute — January 26, 2018

Outline

1. We Need Mathematics to Study Privacy? Seriously?
2. Differential Privacy: Definition, Properties and Basic Mechanisms
3. Differentially Private Machine Learning: ERM and Bayesian Learning
4. Variations on Differential Privacy: Concentrated DP and Local DP
5. Final Remarks

Outline

1. We Need Mathematics to Study Privacy? Seriously?
2. Differential Privacy: Definition, Properties and Basic Mechanisms
3. Differentially Private Machine Learning: ERM and Bayesian Learning
4. Variations on Differential Privacy: Concentrated DP and Local DP
5. Final Remarks

Anonymization Fiascos

Disturbing Headlines and Paper Titles

- ▶ “A Face Is Exposed for AOL Searcher No. 4417749” [Barbaro & Zeller '06]
- ▶ “Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset)” [Narayanan & Shmatikov '08]
- ▶ “Matching Known Patients to Health Records in Washington State Data” [Sweeney '13]
- ▶ “Harvard Professor Re-Identifies Anonymous Volunteers In DNA Study” [Sweeney et al. '13]
- ▶ ... and many others

In general, removing identifiers and applying anonymization heuristics is not always enough!

Why is Anonymization Hard?

- High-dimensional/high-resolution data is essentially unique:

<i>office</i>	<i>department</i>	<i>date joined</i>	<i>salary</i>	<i>d.o.b.</i>	<i>nationality</i>	<i>gender</i>
London	IT	Apr 2015	£####	May 1985	Portuguese	Female

- Lower dimension and lower resolution is more private, but less useful:

<i>office</i>	<i>department</i>	<i>date joined</i>	<i>salary</i>	<i>d.o.b.</i>	<i>nationality</i>	<i>gender</i>
UK	IT	2015	£###	1980-1985	—	Female

Why is Anonymization Hard?

- High-dimensional/high-resolution data is essentially unique:

<i>office</i>	<i>department</i>	<i>date joined</i>	<i>salary</i>	<i>d.o.b.</i>	<i>nationality</i>	<i>gender</i>
London	IT	Apr 2015	£####	May 1985	Portuguese	Female

- Lower dimension and lower resolution is more private, but less useful:

<i>office</i>	<i>department</i>	<i>date joined</i>	<i>salary</i>	<i>d.o.b.</i>	<i>nationality</i>	<i>gender</i>
UK	IT	2015	£###	1980-1985	—	Female

Managing Expectations

Unreasonable Privacy Expectations

- ▶ *Privacy for free?* No, privatizing requires removing information (\Rightarrow accuracy loss)
- ▶ *Absolute privacy?* No, your neighbour's habits are correlated with your habits

Reasonable Privacy Expectations

- ▶ *Quantitative:* offer a knob to tune accuracy vs. privacy loss
- ▶ *Plausible deniability:* your presence in a database cannot be ascertained
- ▶ *Prevent targeted attacks:* limit information leaked even in the presence of side knowledge

The Promise of Differential Privacy

Quote from [Dwork and Roth, 2014]:

Differential privacy describes a promise, made by a data holder, or curator, to a data subject: “You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.”

Quotes from the 2017 Gödel Prize citation awarded to Dwork, McSherry, Nissim and Smith:

Differential privacy was carefully constructed to avoid numerous and subtle pitfalls that other attempts at defining privacy have faced.

The intellectual impact of differential privacy has been broad, with influence on the thinking about privacy being noticeable in a huge range of disciplines, ranging from traditional areas of computer science (databases, machine learning, networking, security) to economics and game theory, false discovery control, official statistics and econometrics, information theory, genomics and, recently, law and policy.

Outline

1. We Need Mathematics to Study Privacy? Seriously?
2. Differential Privacy: Definition, Properties and Basic Mechanisms
3. Differentially Private Machine Learning: ERM and Bayesian Learning
4. Variations on Differential Privacy: Concentrated DP and Local DP
5. Final Remarks

Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with σ -algebra of measurable events)
- ▶ Privacy parameter $\varepsilon \geq 0$

Differential Privacy [Dwork et al., 2006, Dwork, 2006]

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is ε -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E]$$

Intuitions behind the definition:

- ▶ The neighbouring relation \simeq captures *what* is protected
- ▶ The probability bounds capture *how much* protection we get

Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with σ -algebra of measurable events)
- ▶ Privacy parameter $\varepsilon \geq 0$

Differential Privacy [Dwork et al., 2006, Dwork, 2006]

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is ε -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E]$$

Intuitions behind the definition:

- ▶ The neighbouring relation \simeq captures *what* is protected
- ▶ The probability bounds capture *how much* protection we get

Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with σ -algebra of measurable events)
- ▶ Privacy parameter $\varepsilon \geq 0$

Differential Privacy [Dwork et al., 2006, Dwork, 2006]

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is ε -differentially private if **for all** neighbouring inputs $x \simeq x'$ and **for all** sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E]$$

Intuitions behind the definition:

- ▶ The neighbouring relation \simeq captures *what* is protected
- ▶ The probability bounds capture *how much* protection we get

Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with σ -algebra of measurable events)
- ▶ Privacy parameter $\varepsilon \geq 0$

Differential Privacy [Dwork et al., 2006, Dwork, 2006]

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is ε -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E]$$

Intuitions behind the definition:

- ▶ The neighbouring relation \simeq captures *what* is protected
- ▶ The probability bounds capture *how much* protection we get

Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with σ -algebra of measurable events)
- ▶ Privacy parameter $\varepsilon \geq 0$

Differential Privacy [Dwork et al., 2006, Dwork, 2006]

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is ε -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E]$$

Intuitions behind the definition:

- ▶ The neighbouring relation \simeq captures *what* is protected
- ▶ The probability bounds capture *how much* protection we get

Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with σ -algebra of measurable events)
- ▶ Privacy parameter $\varepsilon \geq 0$

Differential Privacy [Dwork et al., 2006, Dwork, 2006]

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is ε -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E]$$

Intuitions behind the definition:

- ▶ The neighbouring relation \simeq captures *what* is protected
- ▶ The probability bounds capture *how much* protection we get

Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with σ -algebra of measurable events)
- ▶ Privacy parameter $\varepsilon \geq 0$

Differential Privacy [Dwork et al., 2006, Dwork, 2006]

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is ε -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E]$$

Intuitions behind the definition:

- ▶ The neighbouring relation \simeq captures *what* is protected
- ▶ The probability bounds capture *how much* protection we get

DP before DP: Randomized Response

The Randomized Response Mechanism [Warner, 1965]

- n individuals answer a survey with one binary question
- The truthful answer for individual i is $x_i \in \{0, 1\}$
- Each individual answers truthfully ($y_i = x_i$) with probability $e^\varepsilon / (1 + e^\varepsilon)$ and falsely ($y_i = \bar{x}_i$) with probability $1 / (1 + e^\varepsilon)$
- Let's denote the mechanism by $(y_1, \dots, y_n) = RR_\varepsilon(x_1, \dots, x_n)$

Intuition: Provides plausible deniability for each individual's answer

Claim: RR_ε is ε -DP (*free-range organic proof on the whiteboard*)

Utility: Averaging the (unbiased) answers \tilde{y}_i from RR_ε satisfies w.h.p.

$$\left| \frac{1}{n} \sum_{i=1}^n x_i - \frac{1}{n} \sum_{i=1}^n \tilde{y}_i \right| \leq \mathcal{O}\left(\frac{1}{\varepsilon \sqrt{n}}\right)$$

DP before DP: Randomized Response

The Randomized Response Mechanism [Warner, 1965]

- n individuals answer a survey with one binary question
- The truthful answer for individual i is $x_i \in \{0, 1\}$
- Each individual answers truthfully ($y_i = x_i$) with probability $e^\varepsilon / (1 + e^\varepsilon)$ and falsely ($y_i = \bar{x}_i$) with probability $1 / (1 + e^\varepsilon)$
- Let's denote the mechanism by $(y_1, \dots, y_n) = RR_\varepsilon(x_1, \dots, x_n)$

Intuition: Provides plausible deniability for each individual's answer

Claim: RR_ε is ε -DP (*free-range organic proof on the whiteboard*)

Utility: Averaging the (unbiased) answers \tilde{y}_i from RR_ε satisfies w.h.p.

$$\left| \frac{1}{n} \sum_{i=1}^n x_i - \frac{1}{n} \sum_{i=1}^n \tilde{y}_i \right| \leq \mathcal{O}\left(\frac{1}{\varepsilon \sqrt{n}}\right)$$

DP before DP: Randomized Response

The Randomized Response Mechanism [Warner, 1965]

- n individuals answer a survey with one binary question
- The truthful answer for individual i is $x_i \in \{0, 1\}$
- Each individual answers truthfully ($y_i = x_i$) with probability $e^\varepsilon / (1 + e^\varepsilon)$ and falsely ($y_i = \bar{x}_i$) with probability $1 / (1 + e^\varepsilon)$
- Let's denote the mechanism by $(y_1, \dots, y_n) = RR_\varepsilon(x_1, \dots, x_n)$

Intuition: Provides plausible deniability for each individual's answer

Claim: RR_ε is ε -DP (*free-range organic proof on the whiteboard*)

Utility: Averaging the (unbiased) answers \tilde{y}_i from RR_ε satisfies w.h.p.

$$\left| \frac{1}{n} \sum_{i=1}^n x_i - \frac{1}{n} \sum_{i=1}^n \tilde{y}_i \right| \leq \mathcal{O}\left(\frac{1}{\varepsilon \sqrt{n}}\right)$$

DP before DP: Randomized Response

The Randomized Response Mechanism [Warner, 1965]

- n individuals answer a survey with one binary question
- The truthful answer for individual i is $x_i \in \{0, 1\}$
- Each individual answers truthfully ($y_i = x_i$) with probability $e^\varepsilon / (1 + e^\varepsilon)$ and falsely ($y_i = \bar{x}_i$) with probability $1 / (1 + e^\varepsilon)$
- Let's denote the mechanism by $(y_1, \dots, y_n) = RR_\varepsilon(x_1, \dots, x_n)$

Intuition: Provides plausible deniability for each individual's answer

Claim: RR_ε is ε -DP (*free-range organic proof on the whiteboard*)

Utility: Averaging the (unbiased) answers \tilde{y}_i from RR_ε satisfies w.h.p.

$$\left| \frac{1}{n} \sum_{i=1}^n x_i - \frac{1}{n} \sum_{i=1}^n \tilde{y}_i \right| \leq \mathcal{O}\left(\frac{1}{\varepsilon \sqrt{n}}\right)$$

The Laplace Mechanism (for computing the mean)

Private Mean Computation

- ▶ A curator holds one bit $x_i \in \{0, 1\}$ for each of n individuals
- ▶ The curator proceeds by
 1. Computing the mean $\mu = \frac{1}{n} \sum_{i=1}^n x_i$,
 2. Sampling noise $Z \sim \text{Lap}(\frac{1}{\varepsilon n})$, and
 3. Revealing the noisy mean $\tilde{\mu} = \mu + Z$
- ▶ Let's denote the mechanism by $\tilde{\mu} = \mathcal{M}_{\text{Lap}}(x_1, \dots, x_n)$

Claim: \mathcal{M}_{Lap} is ε -DP (*free-range organic proof on the whiteboard*)

Utility: The answer returned by the mechanism satisfies w.h.p.

$$|\mu - \tilde{\mu}| \leq \mathcal{O}\left(\frac{1}{\varepsilon n}\right)$$

The Laplace Mechanism (for computing the mean)

Private Mean Computation

- A curator holds one bit $x_i \in \{0, 1\}$ for each of n individuals
- The curator proceeds by
 1. Computing the mean $\mu = \frac{1}{n} \sum_{i=1}^n x_i$,
 2. Sampling noise $Z \sim \text{Lap}(\frac{1}{\varepsilon n})$, and
 3. Revealing the noisy mean $\tilde{\mu} = \mu + Z$
- Let's denote the mechanism by $\tilde{\mu} = \mathcal{M}_{\text{Lap}}(x_1, \dots, x_n)$

Claim: \mathcal{M}_{Lap} is ε -DP (*free-range organic proof on the whiteboard*)

Utility: The answer returned by the mechanism satisfies w.h.p.

$$|\mu - \tilde{\mu}| \leq \mathcal{O}\left(\frac{1}{\varepsilon n}\right)$$

The Laplace Mechanism (for computing the mean)

Private Mean Computation

- ▶ A curator holds one bit $x_i \in \{0, 1\}$ for each of n individuals
- ▶ The curator proceeds by
 1. Computing the mean $\mu = \frac{1}{n} \sum_{i=1}^n x_i$,
 2. Sampling noise $Z \sim \text{Lap}(\frac{1}{\varepsilon n})$, and
 3. Revealing the noisy mean $\tilde{\mu} = \mu + Z$
- ▶ Let's denote the mechanism by $\tilde{\mu} = \mathcal{M}_{\text{Lap}}(x_1, \dots, x_n)$

Claim: \mathcal{M}_{Lap} is ε -DP (*free-range organic proof on the whiteboard*)

Utility: The answer returned by the mechanism satisfies w.h.p.

$$|\mu - \tilde{\mu}| \leq \mathcal{O}\left(\frac{1}{\varepsilon n}\right)$$

Approximate Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with sigma-algebra of measurable events)
- ▶ Privacy parameters $\varepsilon \geq 0$, $\delta \in [0, 1]$

Approximate Differential Privacy

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is (ε, δ) -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E] + \delta$$

Interpretation

- ▶ δ accounts for “bad events” that might result in high privacy losses
- ▶ Mechanism $\mathcal{M}(x_1, \dots, x_n) = x_{\text{Unif}([n])}$ is $(0, 1/n)$ -DP (\Rightarrow should take $\delta \ll 1/n$)

Approximate Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with sigma-algebra of measurable events)
- ▶ Privacy parameters $\varepsilon \geq 0$, $\delta \in [0, 1]$

Approximate Differential Privacy

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is (ε, δ) -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E] + \delta$$

Interpretation

- ▶ δ accounts for “bad events” that might result in high privacy losses
- ▶ Mechanism $\mathcal{M}(x_1, \dots, x_n) = x_{\text{Unif}([n])}$ is $(0, 1/n)$ -DP (\Rightarrow should take $\delta \ll 1/n$)

Approximate Differential Privacy

Ingredients

- ▶ Input space X (with symmetric neighbouring relation \simeq)
- ▶ Output space Y (with sigma-algebra of measurable events)
- ▶ Privacy parameters $\varepsilon \geq 0$, $\delta \in [0, 1]$

Approximate Differential Privacy

A randomized mechanism $\mathcal{M} : X \rightarrow Y$ is (ε, δ) -differentially private if for all neighbouring inputs $x \simeq x'$ and for all sets of outputs $E \subseteq Y$ we have

$$\mathbb{P}[\mathcal{M}(x) \in E] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(x') \in E] + \delta$$

Interpretation

- ▶ δ accounts for “bad events” that might result in high privacy losses
- ▶ Mechanism $\mathcal{M}(x_1, \dots, x_n) = x_{\text{Unif}([n])}$ is $(0, 1/n)$ -DP (\Rightarrow should take $\delta \ll 1/n$)



Output Perturbation Mechanisms

The Laplace mechanism is an example of a more general class of mechanisms

Global Sensitivity: for any function $f : X \rightarrow \mathbb{R}^d$ define $\Delta_p = \sup_{x \simeq x'} \|f(x) - f(x')\|_p$

Output Perturbation (with Laplace and Gaussian noise)

- ▶ A curator holds one vector $x_i \in \mathbb{R}^d$ for each of n individuals
- ▶ The curator computes a function $f(x_1, \dots, x_n)$ of the data,
- ▶ samples noise $Z \sim \text{Lap}(\frac{\Delta_1}{\epsilon})^d$ or $Z \sim \mathcal{N}(0, \sigma^2)^d$ with $\sigma = \frac{\Delta_2 \sqrt{C \log(1/\delta)}}{\epsilon}$, and
- ▶ reveals the noisy value $f(x_1, \dots, x_n) + Z$
- ▶ Let's denote the mechanisms $\mathcal{M}_{f, \text{Lap}}$ and $\mathcal{M}_{f, \mathcal{N}}$ respectively
- ▶ Note the mechanism of the previous slide is $\mathcal{M}_{f, \text{Lap}}$ for $f(x_1, \dots, x_n) = \frac{1}{n} \sum_{i=1}^n x_i$

Claim: $\mathcal{M}_{f, \text{Lap}}$ is ϵ -DP and $\mathcal{M}_{f, \mathcal{N}}$ is (ϵ, δ) -DP

Output Perturbation Mechanisms

The Laplace mechanism is an example of a more general class of mechanisms

Global Sensitivity: for any function $f : X \rightarrow \mathbb{R}^d$ define $\Delta_p = \sup_{x \simeq x'} \|f(x) - f(x')\|_p$

Output Perturbation (with Laplace and Gaussian noise)

- ▶ A curator holds one vector $x_i \in \mathbb{R}^d$ for each of n individuals
- ▶ The curator computes a function $f(x_1, \dots, x_n)$ of the data,
- ▶ samples noise $Z \sim \text{Lap}(\frac{\Delta_1}{\epsilon})^d$ or $Z \sim \mathcal{N}(0, \sigma^2)^d$ with $\sigma = \frac{\Delta_2 \sqrt{C \log(1/\delta)}}{\epsilon}$, and
- ▶ reveals the noisy value $f(x_1, \dots, x_n) + Z$
- ▶ Let's denote the mechanisms $\mathcal{M}_{f, \text{Lap}}$ and $\mathcal{M}_{f, \mathcal{N}}$ respectively
- ▶ Note the mechanism of the previous slide is $\mathcal{M}_{f, \text{Lap}}$ for $f(x_1, \dots, x_n) = \frac{1}{n} \sum_{i=1}^n x_i$

Claim: $\mathcal{M}_{f, \text{Lap}}$ is ϵ -DP and $\mathcal{M}_{f, \mathcal{N}}$ is (ϵ, δ) -DP

Output Perturbation Mechanisms

The Laplace mechanism is an example of a more general class of mechanisms

Global Sensitivity: for any function $f : X \rightarrow \mathbb{R}^d$ define $\Delta_p = \sup_{x \simeq x'} \|f(x) - f(x')\|_p$

Output Perturbation (with Laplace and Gaussian noise)

- ▶ A curator holds one vector $x_i \in \mathbb{R}^d$ for each of n individuals
- ▶ The curator computes a function $f(x_1, \dots, x_n)$ of the data,
- ▶ samples noise $Z \sim \text{Lap}(\frac{\Delta_1}{\epsilon})^d$ or $Z \sim \mathcal{N}(0, \sigma^2)^d$ with $\sigma = \frac{\Delta_2 \sqrt{C \log(1/\delta)}}{\epsilon}$, and
- ▶ reveals the noisy value $f(x_1, \dots, x_n) + Z$
- ▶ Let's denote the mechanisms $\mathcal{M}_{f, \text{Lap}}$ and $\mathcal{M}_{f, \mathcal{N}}$ respectively
- ▶ Note the mechanism of the previous slide is $\mathcal{M}_{f, \text{Lap}}$ for $f(x_1, \dots, x_n) = \frac{1}{n} \sum_{i=1}^n x_i$

Claim: $\mathcal{M}_{f, \text{Lap}}$ is ϵ -DP and $\mathcal{M}_{f, \mathcal{N}}$ is (ϵ, δ) -DP

Fundamental Properties

- Robustness to post-processing: \mathcal{M} is (ε, δ) -DP, then $F \circ \mathcal{M}$ is (ε, δ) -DP
- Composition: if \mathcal{M}_j , $j = 1, \dots, k$, are $(\varepsilon_j, \delta_j)$ -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\sum_j \varepsilon_j, \sum_j \delta_j)$ -DP. In the homogeneous case this yields $(k\varepsilon, k\delta)$ -DP
- Advanced composition: if \mathcal{M}_j , $j = 1, \dots, k$, are (ε, δ) -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\varepsilon\sqrt{k \log(1/\delta')} + \varepsilon(e^\varepsilon - 1)k, k\delta + \delta')$ -DP for any $\delta' > 0$
- Group privacy: if \mathcal{M} is (ε, δ) -DP with respect to $x \simeq x'$, then \mathcal{M} is $(t\varepsilon, te^{t\varepsilon}\delta)$ with respect to $x \simeq^t x'$ (ie. t changes)
- Protects against side knowledge: if attacker has prior $P_{prior}^{x_i}$ and computes $P_{posterior}^{x_i}$ after observing $\mathcal{M}(\vec{x})$ from ε -DP mechanism, then $\text{dist}(P_{prior}^{x_i}, P_{posterior}^{x_i}) = \mathcal{O}(\varepsilon)$

Fundamental Properties

- Robustness to post-processing: \mathcal{M} is (ε, δ) -DP, then $F \circ \mathcal{M}$ is (ε, δ) -DP
- Composition: if \mathcal{M}_j , $j = 1, \dots, k$, are $(\varepsilon_j, \delta_j)$ -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\sum_j \varepsilon_j, \sum_j \delta_j)$ -DP. In the homogeneous case this yields $(k\varepsilon, k\delta)$ -DP
- Advanced composition: if \mathcal{M}_j , $j = 1, \dots, k$, are (ε, δ) -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\varepsilon\sqrt{k \log(1/\delta')} + \varepsilon(e^\varepsilon - 1)k, k\delta + \delta')$ -DP for any $\delta' > 0$
- Group privacy: if \mathcal{M} is (ε, δ) -DP with respect to $x \simeq x'$, then \mathcal{M} is $(t\varepsilon, te^{t\varepsilon}\delta)$ with respect to $x \simeq^t x'$ (ie. t changes)
- Protects against side knowledge: if attacker has prior $P_{prior}^{x_i}$ and computes $P_{posterior}^{x_i}$ after observing $\mathcal{M}(\vec{x})$ from ε -DP mechanism, then $\text{dist}(P_{prior}^{x_i}, P_{posterior}^{x_i}) = \mathcal{O}(\varepsilon)$

Fundamental Properties

- Robustness to post-processing: \mathcal{M} is (ε, δ) -DP, then $F \circ \mathcal{M}$ is (ε, δ) -DP
- Composition: if \mathcal{M}_j , $j = 1, \dots, k$, are $(\varepsilon_j, \delta_j)$ -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\sum_j \varepsilon_j, \sum_j \delta_j)$ -DP. In the homogeneous case this yields $(k\varepsilon, k\delta)$ -DP
- Advanced composition: if \mathcal{M}_j , $j = 1, \dots, k$, are (ε, δ) -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\varepsilon\sqrt{k \log(1/\delta')} + \varepsilon(e^\varepsilon - 1)k, k\delta + \delta')$ -DP for any $\delta' > 0$
- Group privacy: if \mathcal{M} is (ε, δ) -DP with respect to $x \simeq x'$, then \mathcal{M} is $(t\varepsilon, te^{t\varepsilon}\delta)$ with respect to $x \simeq^t x'$ (ie. t changes)
- Protects against side knowledge: if attacker has prior $P_{prior}^{x_i}$ and computes $P_{posterior}^{x_i}$ after observing $\mathcal{M}(\vec{x})$ from ε -DP mechanism, then $\text{dist}(P_{prior}^{x_i}, P_{posterior}^{x_i}) = \mathcal{O}(\varepsilon)$

Fundamental Properties

- ▶ Robustness to post-processing: \mathcal{M} is (ε, δ) -DP, then $F \circ \mathcal{M}$ is (ε, δ) -DP
- ▶ Composition: if \mathcal{M}_j , $j = 1, \dots, k$, are $(\varepsilon_j, \delta_j)$ -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\sum_j \varepsilon_j, \sum_j \delta_j)$ -DP. In the homogeneous case this yields $(k\varepsilon, k\delta)$ -DP
- ▶ Advanced composition: if \mathcal{M}_j , $j = 1, \dots, k$, are (ε, δ) -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\varepsilon\sqrt{k \log(1/\delta')} + \varepsilon(e^\varepsilon - 1)k, k\delta + \delta')$ -DP for any $\delta' > 0$
- ▶ Group privacy: if \mathcal{M} is (ε, δ) -DP with respect to $x \simeq x'$, then \mathcal{M} is $(t\varepsilon, te^{t\varepsilon}\delta)$ with respect to $x \simeq^t x'$ (ie. t changes)
- ▶ Protects against side knowledge: if attacker has prior $P_{prior}^{x_i}$ and computes $P_{posterior}^{x_i}$ after observing $\mathcal{M}(\vec{x})$ from ε -DP mechanism, then $\text{dist}(P_{prior}^{x_i}, P_{posterior}^{x_i}) = \mathcal{O}(\varepsilon)$

Fundamental Properties

- Robustness to post-processing: \mathcal{M} is (ε, δ) -DP, then $F \circ \mathcal{M}$ is (ε, δ) -DP
- Composition: if \mathcal{M}_j , $j = 1, \dots, k$, are $(\varepsilon_j, \delta_j)$ -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\sum_j \varepsilon_j, \sum_j \delta_j)$ -DP. In the homogeneous case this yields $(k\varepsilon, k\delta)$ -DP
- Advanced composition: if \mathcal{M}_j , $j = 1, \dots, k$, are (ε, δ) -DP, then $\vec{x} \mapsto (\mathcal{M}_1(\vec{x}), \dots, \mathcal{M}_k(\vec{x}))$ is $(\varepsilon\sqrt{k \log(1/\delta')} + \varepsilon(e^\varepsilon - 1)k, k\delta + \delta')$ -DP for any $\delta' > 0$
- Group privacy: if \mathcal{M} is (ε, δ) -DP with respect to $x \simeq x'$, then \mathcal{M} is $(t\varepsilon, te^{t\varepsilon}\delta)$ with respect to $x \simeq^t x'$ (ie. t changes)
- Protects against side knowledge: if attacker has prior $P_{prior}^{x_i}$ and computes $P_{posterior}^{x_i}$ after observing $\mathcal{M}(\vec{x})$ from ε -DP mechanism, then $\text{dist}(P_{prior}^{x_i}, P_{posterior}^{x_i}) = \mathcal{O}(\varepsilon)$

The Exponential Mechanism

The Laplace and Gaussian mechanisms are examples of a more general class of mechanisms

Densities of output perturbation mechanisms

$$p_{\mathcal{M}_{f,\text{Lap}}(x)}(y) \propto \exp\left(\frac{-\varepsilon \|y - f(x)\|_1}{\Delta_1}\right) \quad p_{\mathcal{M}_{f,\mathcal{N}}(x)}(y) \propto \exp\left(\frac{-\varepsilon^2 \|y - f(x)\|_2^2}{C\Delta_2^2 \log(1/\delta)}\right)$$

Exponential Mechanism

- ▶ Prior distribution over outputs with density π
- ▶ Scoring function $q : X \times Y \rightarrow \mathbb{R}_{\geq 0}$ provides scores for each output y w.r.t. input x
- ▶ The exponential mechanism $\mathcal{M}_{\pi,q}(x)$ outputs a sample from the distribution with density

$$p_{\pi,q}(y) \propto \pi(y) \exp(-\beta q(x, y))$$

The Exponential Mechanism

The Laplace and Gaussian mechanisms are examples of a more general class of mechanisms

Densities of output perturbation mechanisms

$$p_{\mathcal{M}_{f,\text{Lap}}(x)}(y) \propto \exp\left(\frac{-\varepsilon \|y - f(x)\|_1}{\Delta_1}\right) \quad p_{\mathcal{M}_{f,\mathcal{N}}(x)}(y) \propto \exp\left(\frac{-\varepsilon^2 \|y - f(x)\|_2^2}{C\Delta_2^2 \log(1/\delta)}\right)$$

Exponential Mechanism

- ▶ Prior distribution over outputs with density π
- ▶ Scoring function $q : X \times Y \rightarrow \mathbb{R}_{\geq 0}$ provides scores for each output y w.r.t. input x
- ▶ The exponential mechanism $\mathcal{M}_{\pi,q}(x)$ outputs a sample from the distribution with density

$$p_{\pi,q}(y) \propto \pi(y) \exp(-\beta q(x, y))$$

The Exponential Mechanism

The Laplace and Gaussian mechanisms are examples of a more general class of mechanisms

Densities of output perturbation mechanisms

$$p_{\mathcal{M}_{f,\text{Lap}}(x)}(y) \propto \exp\left(\frac{-\varepsilon \|y - f(x)\|_1}{\Delta_1}\right) \quad p_{\mathcal{M}_{f,\mathcal{N}}(x)}(y) \propto \exp\left(\frac{-\varepsilon^2 \|y - f(x)\|_2^2}{C\Delta_2^2 \log(1/\delta)}\right)$$

Exponential Mechanism

- ▶ Prior distribution over outputs with density π
- ▶ Scoring function $q : X \times Y \rightarrow \mathbb{R}_{\geq 0}$ provides scores for each output y w.r.t. input x
- ▶ The exponential mechanism $\mathcal{M}_{\pi,q}(x)$ outputs a sample from the distribution with density

$$p_{\pi,q}(y) \propto \pi(y) \exp(-\beta q(x, y))$$

Calibrating The Exponential Mechanism

Properties of the Scoring Function

- Sensitivity: $\sup_{x \simeq x'} \sup_y |q(x, y) - q(x', y)| \leq \Delta$
- Lipschitz: $\sup_{x \simeq x'} |(q(x, y) - q(x', y)) - (q(x, y') - q(x', y'))| \leq L \|y - y'\|$

Properties of the Prior

- Strong log-concavity: $\pi(y) = e^{-W(y)}$ for some κ -strongly convex W

Privacy Guarantees for the Exponential Mechanism

Assumptions	β	Privacy	Reference
q bounded sensitivity	$\mathcal{O}\left(\frac{\varepsilon}{\Delta}\right)$	$(\varepsilon, 0)$	[McSherry and Talwar, 2007]
q Lipschitz + convex π strongly log-concave	$\mathcal{O}\left(\frac{\varepsilon\sqrt{\kappa}}{L\sqrt{\log(1/\delta)}}\right)$	(ε, δ)	[Minami et al., 2016]

Outline

1. We Need Mathematics to Study Privacy? Seriously?
2. Differential Privacy: Definition, Properties and Basic Mechanisms
3. Differentially Private Machine Learning: ERM and Bayesian Learning
4. Variations on Differential Privacy: Concentrated DP and Local DP
5. Final Remarks

Differentially Private Empirical Risk Minimization

Setup: A curator has features and labels $\vec{z} = ((x_1, y_1), \dots, (x_n, y_n))$ about n individuals and wants to train a model by minimizing over $\theta \in \Theta$

$$L(\vec{z}, \theta) = \frac{1}{n} \sum_{i=1}^n \ell(x_i, y_i, \theta) + \frac{R(\theta)}{n}$$

Examples: logistic regression, SVM, linear regression, DNN, etc.

Private ERM Algorithms

- ▶ Output Perturbation: add some noise Z to $\hat{\theta} = \operatorname{argmin}_{\theta \in \Theta} L(\vec{z}, \theta)$
- ▶ Objective Perturbation: reveal the optimum of $L(\vec{z}, \theta) + \langle \theta, Z \rangle$ for some noise Z
- ▶ Gradient Perturbation: optimize $L(\vec{z}, \theta)$ using mini-batch SGD with noisy gradients

Differentially Private Empirical Risk Minimization

Setup: A curator has features and labels $\vec{z} = ((x_1, y_1), \dots, (x_n, y_n))$ about n individuals and wants to train a model by minimizing over $\theta \in \Theta$

$$L(\vec{z}, \theta) = \frac{1}{n} \sum_{i=1}^n \ell(x_i, y_i, \theta) + \frac{R(\theta)}{n}$$

Examples: logistic regression, SVM, linear regression, DNN, etc.

Private ERM Algorithms

- ▶ Output Perturbation: add some noise Z to $\hat{\theta} = \operatorname{argmin}_{\theta \in \Theta} L(\vec{z}, \theta)$
- ▶ Objective Perturbation: reveal the optimum of $L(\vec{z}, \theta) + \langle \theta, Z \rangle$ for some noise Z
- ▶ Gradient Perturbation: optimize $L(\vec{z}, \theta)$ using mini-batch SGD with noisy gradients

DP-ERM: Method Comparison

Perturb	Optimization	Privacy	Assumptions	Excess Risk	Reference
Objective	Exact	(ε, δ)	linear model convexity	$\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon\sqrt{n}}\right)$	[Jain and Thakurta, 2014]
Output	Exact	(ε, δ)	linear model convexity	$\mathcal{O}\left(\frac{1}{\varepsilon\sqrt{n}}\right)$	[Jain and Thakurta, 2014]
Output	SGD	ε	linear model convexity	$\mathcal{O}\left(\frac{d}{\varepsilon\sqrt{n}}\right)$	[Wu et al., 2016]
Output	SGD	ε	linear model strong convexity	$\mathcal{O}\left(\frac{d}{\varepsilon n}\right)$	[Wu et al., 2016]
Gradient	SGD	(ε, δ)	convexity	$\tilde{\mathcal{O}}\left(\frac{\sqrt{d}}{\varepsilon n}\right)$	[Bassily et al., 2014]
Gradient	SGD	(ε, δ)	strong convexity	$\tilde{\mathcal{O}}\left(\frac{d}{\varepsilon^2 n^2}\right)$	[Bassily et al., 2014]

See also [Talwar et al., 2014, Abadi et al., 2016]

Private Bayesian Learning

One-Posterior Sample (OPS) Mechanism [Wang et al., 2015]

- Curator has a prior $P_{prior}(\theta)$ and a model $P_{model}(x_i|\theta)$
- Given a dataset \vec{x} the curators computes the posterior $P_{posterior}(\theta|\vec{x})$, and
- reveals a sample $\hat{\theta} \sim P_{posterior}(\theta|\vec{x})$

Claim: If the model satisfies $\sup_{x,x',\theta} |\log P_{model}(x|\theta) - \log P_{model}(x'|\theta)| \leq \varepsilon/2$ then OPS is ε -DP

See also: [Wang et al., 2015, Foulds et al., 2016, Minami et al., 2016] for DP with approximate inference, [Park et al., 2016] for DP with variational Bayes, and [Zhang et al., 2016] for Bayesian network mechanisms

Private Bayesian Learning

One-Posterior Sample (OPS) Mechanism [Wang et al., 2015]

- Curator has a prior $P_{prior}(\theta)$ and a model $P_{model}(x_i|\theta)$
- Given a dataset \vec{x} the curators computes the posterior $P_{posterior}(\theta|\vec{x})$, and
- reveals a sample $\hat{\theta} \sim P_{posterior}(\theta|\vec{x})$

Claim: If the model satisfies $\sup_{x,x',\theta} |\log P_{model}(x|\theta) - \log P_{model}(x'|\theta)| \leq \varepsilon/2$ then OPS is ε -DP

See also: [Wang et al., 2015, Foulds et al., 2016, Minami et al., 2016] for DP with approximate inference, [Park et al., 2016] for DP with variational Bayes, and [Zhang et al., 2016] for Bayesian network mechanisms

Private Bayesian Learning

One-Posterior Sample (OPS) Mechanism [Wang et al., 2015]

- Curator has a prior $P_{prior}(\theta)$ and a model $P_{model}(x_i|\theta)$
- Given a dataset \vec{x} the curators computes the posterior $P_{posterior}(\theta|\vec{x})$, and
- reveals a sample $\hat{\theta} \sim P_{posterior}(\theta|\vec{x})$

Claim: If the model satisfies $\sup_{x,x',\theta} |\log P_{model}(x|\theta) - \log P_{model}(x'|\theta)| \leq \varepsilon/2$ then OPS is ε -DP

See also: [Wang et al., 2015, Foulds et al., 2016, Minami et al., 2016] for DP with approximate inference, [Park et al., 2016] for DP with variational Bayes, and [Zhang et al., 2016] for Bayesian network mechanisms

Outline

1. We Need Mathematics to Study Privacy? Seriously?
2. Differential Privacy: Definition, Properties and Basic Mechanisms
3. Differentially Private Machine Learning: ERM and Bayesian Learning
4. Variations on Differential Privacy: Concentrated DP and Local DP
5. Final Remarks

Privacy Losses

Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with density function $p_{\mathcal{M}(x)}(y)$

Privacy Loss (function)

$$\mathcal{L}_{\mathcal{M}, x, x'}(y) = \log \left(\frac{p_{\mathcal{M}(x)}(y)}{p_{\mathcal{M}(x')}(y)} \right)$$

Privacy Loss (random variable)

$$L_{\mathcal{M}, x, x'} = \mathcal{L}_{\mathcal{M}, x, x'}(\mathcal{M}(x))$$

Lemma (Sufficient Condition)

A mechanism $\mathcal{M} : X \rightarrow Y$ is (ϵ, δ) -DP if for any $x \simeq x'$ we have $\mathbb{P}[L_{\mathcal{M}, x, x'} \geq \epsilon] \leq \delta$

Privacy Losses

Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with density function $p_{\mathcal{M}(x)}(y)$

Privacy Loss (function)

$$\mathcal{L}_{\mathcal{M}, x, x'}(y) = \log \left(\frac{p_{\mathcal{M}(x)}(y)}{p_{\mathcal{M}(x')}(y)} \right)$$

Privacy Loss (random variable)

$$L_{\mathcal{M}, x, x'} = \mathcal{L}_{\mathcal{M}, x, x'}(\mathcal{M}(x))$$

Lemma (Sufficient Condition)

A mechanism $\mathcal{M} : X \rightarrow Y$ is (ϵ, δ) -DP if for any $x \simeq x'$ we have $\mathbb{P}[L_{\mathcal{M}, x, x'} \geq \epsilon] \leq \delta$

Privacy Losses

Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with density function $p_{\mathcal{M}(x)}(y)$

Privacy Loss (function)

$$\mathcal{L}_{\mathcal{M},x,x'}(y) = \log \left(\frac{p_{\mathcal{M}(x)}(y)}{p_{\mathcal{M}(x')}(y)} \right)$$

Privacy Loss (random variable)

$$L_{\mathcal{M},x,x'} = \mathcal{L}_{\mathcal{M},x,x'}(\mathcal{M}(x))$$

Lemma (Sufficient Condition)

A mechanism $\mathcal{M} : X \rightarrow Y$ is (ε, δ) -DP if for any $x \simeq x'$ we have $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$

Privacy Losses

Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with density function $p_{\mathcal{M}(x)}(y)$

Privacy Loss (function)

$$\mathcal{L}_{\mathcal{M},x,x'}(y) = \log \left(\frac{p_{\mathcal{M}(x)}(y)}{p_{\mathcal{M}(x')}(y)} \right)$$

Privacy Loss (random variable)

$$L_{\mathcal{M},x,x'} = \mathcal{L}_{\mathcal{M},x,x'}(\mathcal{M}(x))$$

Lemma (Sufficient Condition)

A mechanism $\mathcal{M} : X \rightarrow Y$ is (ε, δ) -DP if for any $x \simeq x'$ we have $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$

Analysis of the Gaussian Mechanism

1. Setup: $\mathcal{M}(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\Delta_2}{\varepsilon} \sqrt{C \log(1/\delta)}$ (for $\varepsilon \leq 1$)
2. Compute the distribution of the privacy loss random variable:

$$\mathcal{L}_{\mathcal{M},x,x'}(y) = \frac{\|y - f(x')\|_2^2 - \|y - f(x)\|_2^2}{2\sigma^2} = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} + \frac{\langle y - f(x), f(x) - f(x') \rangle}{\sigma^2}$$
$$L_{\mathcal{M},x,x'} = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} + \frac{\langle Z, f(x) - f(x') \rangle}{\sigma^2} \sim \mathcal{N}\left(\frac{\|f(x) - f(x')\|_2^2}{2\sigma^2}, \frac{\|f(x) - f(x')\|_2^2}{\sigma^2}\right)$$

3. Use a concentration bound for Gaussian random variables. With probability $\geq 1 - \delta$:

$$\mathcal{N}(\eta, 2\eta) \leq \eta + \sqrt{C_0 \eta \log(1/\delta)} \leq \varepsilon$$

4. Assuming $\varepsilon \leq 1$, a bit of algebra shows $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$ if:

$$\eta \leq \left(\sqrt{\varepsilon + C_1 \log(1/\delta)} - \sqrt{C_1 \log(1/\delta)} \right)^2 \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

5. Substitute the definition of σ^2 and verify the condition is satisfied:

Analysis of the Gaussian Mechanism

1. Setup: $\mathcal{M}(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\Delta_2}{\varepsilon} \sqrt{C \log(1/\delta)}$ (for $\varepsilon \leq 1$)
2. Compute the distribution of the privacy loss random variable:

$$\mathcal{L}_{\mathcal{M},x,x'}(y) = \frac{\|y - f(x')\|_2^2 - \|y - f(x)\|_2^2}{2\sigma^2} = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} + \frac{\langle y - f(x), f(x) - f(x') \rangle}{\sigma^2}$$
$$L_{\mathcal{M},x,x'} = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} + \frac{\langle Z, f(x) - f(x') \rangle}{\sigma^2} \sim \mathcal{N}\left(\frac{\|f(x) - f(x')\|_2^2}{2\sigma^2}, \frac{\|f(x) - f(x')\|_2^2}{\sigma^2}\right)$$

3. Use a concentration bound for Gaussian random variables. With probability $\geq 1 - \delta$:

$$\mathcal{N}(\eta, 2\eta) \leq \eta + \sqrt{C_0 \eta \log(1/\delta)} \leq \varepsilon$$

4. Assuming $\varepsilon \leq 1$, a bit of algebra shows $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$ if:

$$\eta \leq \left(\sqrt{\varepsilon + C_1 \log(1/\delta)} - \sqrt{C_1 \log(1/\delta)} \right)^2 \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

5. Substitute the definition of σ^2 and verify the condition is satisfied:

Analysis of the Gaussian Mechanism

1. Setup: $\mathcal{M}(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\Delta_2}{\varepsilon} \sqrt{C \log(1/\delta)}$ (for $\varepsilon \leq 1$)
2. Compute the distribution of the privacy loss random variable:

$$\mathcal{L}_{\mathcal{M},x,x'}(y) = \frac{\|y - f(x')\|_2^2 - \|y - f(x)\|_2^2}{2\sigma^2} = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} + \frac{\langle y - f(x), f(x) - f(x') \rangle}{\sigma^2}$$
$$L_{\mathcal{M},x,x'} = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} + \frac{\langle Z, f(x) - f(x') \rangle}{\sigma^2} \sim \mathcal{N}\left(\frac{\|f(x) - f(x')\|_2^2}{2\sigma^2}, \frac{\|f(x) - f(x')\|_2^2}{\sigma^2}\right)$$

3. Use a concentration bound for Gaussian random variables. With probability $\geq 1 - \delta$:

$$\mathcal{N}(\eta, 2\eta) \leq \eta + \sqrt{C_0 \eta \log(1/\delta)} \leq \varepsilon$$

4. Assuming $\varepsilon \leq 1$, a bit of algebra shows $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$ if:

$$\eta \leq \left(\sqrt{\varepsilon + C_1 \log(1/\delta)} - \sqrt{C_1 \log(1/\delta)} \right)^2 \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

5. Substitute the definition of σ^2 and verify the condition is satisfied:

Analysis of the Gaussian Mechanism

1. Setup: $\mathcal{M}(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\Delta_2}{\varepsilon} \sqrt{C \log(1/\delta)}$ (for $\varepsilon \leq 1$)
2. Compute the distribution of the privacy loss random variable:

$$L_{\mathcal{M},x,x'} \sim \mathcal{N}\left(\frac{\|f(x) - f(x')\|_2^2}{2\sigma^2}, \frac{\|f(x) - f(x')\|_2^2}{\sigma^2}\right) = \mathcal{N}(\eta, 2\eta)$$

3. Use a concentration bound for Gaussian random variables. With probability $\geq 1 - \delta$:

$$\mathcal{N}(\eta, 2\eta) \leq \eta + \sqrt{C_0 \eta \log(1/\delta)} \leq \varepsilon$$

4. Assuming $\varepsilon \leq 1$, a bit of algebra shows $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$ if:

$$\eta \leq \left(\sqrt{\varepsilon + C_1 \log(1/\delta)} - \sqrt{C_1 \log(1/\delta)} \right)^2 \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

5. Substitute the definition of σ^2 and verify the condition is satisfied:

$$\eta = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} = \frac{\varepsilon^2 \|f(x) - f(x')\|_2^2}{2\Delta_2^2 C \log(1/\delta)} \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

Analysis of the Gaussian Mechanism

1. Setup: $\mathcal{M}(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\Delta_2}{\varepsilon} \sqrt{C \log(1/\delta)}$ (for $\varepsilon \leq 1$)
2. Compute the distribution of the privacy loss random variable:

$$L_{\mathcal{M},x,x'} \sim \mathcal{N}\left(\frac{\|f(x) - f(x')\|_2^2}{2\sigma^2}, \frac{\|f(x) - f(x')\|_2^2}{\sigma^2}\right) = \mathcal{N}(\eta, 2\eta)$$

3. Use a concentration bound for Gaussian random variables. With probability $\geq 1 - \delta$:

$$\mathcal{N}(\eta, 2\eta) \leq \eta + \sqrt{C_0 \eta \log(1/\delta)} \leq \varepsilon$$

4. Assuming $\varepsilon \leq 1$, a bit of algebra shows $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$ if:

$$\eta \leq \left(\sqrt{\varepsilon + C_1 \log(1/\delta)} - \sqrt{C_1 \log(1/\delta)} \right)^2 \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

5. Substitute the definition of σ^2 and verify the condition is satisfied:

$$\eta = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} = \frac{\varepsilon^2 \|f(x) - f(x')\|_2^2}{2\Delta_2^2 C \log(1/\delta)} \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$



Analysis of the Gaussian Mechanism

1. Setup: $\mathcal{M}(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\Delta_2}{\varepsilon} \sqrt{C \log(1/\delta)}$ (for $\varepsilon \leq 1$)
2. Compute the distribution of the privacy loss random variable:

$$L_{\mathcal{M},x,x'} \sim \mathcal{N}\left(\frac{\|f(x) - f(x')\|_2^2}{2\sigma^2}, \frac{\|f(x) - f(x')\|_2^2}{\sigma^2}\right) = \mathcal{N}(\eta, 2\eta)$$

3. Use a concentration bound for Gaussian random variables. With probability $\geq 1 - \delta$:

$$\mathcal{N}(\eta, 2\eta) \leq \eta + \sqrt{C_0 \eta \log(1/\delta)} \leq \varepsilon$$

4. Assuming $\varepsilon \leq 1$, a bit of algebra shows $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$ if:

$$\eta \leq \left(\sqrt{\varepsilon + C_1 \log(1/\delta)} - \sqrt{C_1 \log(1/\delta)} \right)^2 \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

5. Substitute the definition of σ^2 and verify the condition is satisfied:

$$\eta = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} = \frac{\varepsilon^2 \|f(x) - f(x')\|_2^2}{2\Delta_2^2 C \log(1/\delta)} \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

Analysis of the Gaussian Mechanism

1. Setup: $\mathcal{M}(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = \frac{\Delta_2}{\varepsilon} \sqrt{C \log(1/\delta)}$ (for $\varepsilon \leq 1$)
2. Compute the distribution of the privacy loss random variable:

$$L_{\mathcal{M},x,x'} \sim \mathcal{N}\left(\frac{\|f(x) - f(x')\|_2^2}{2\sigma^2}, \frac{\|f(x) - f(x')\|_2^2}{\sigma^2}\right) = \mathcal{N}(\eta, 2\eta)$$

3. Use a concentration bound for Gaussian random variables. With probability $\geq 1 - \delta$:

$$\mathcal{N}(\eta, 2\eta) \leq \eta + \sqrt{C_0 \eta \log(1/\delta)} \leq \varepsilon$$

4. Assuming $\varepsilon \leq 1$, a bit of algebra shows $\mathbb{P}[L_{\mathcal{M},x,x'} \geq \varepsilon] \leq \delta$ if:

$$\eta \leq \left(\sqrt{\varepsilon + C_1 \log(1/\delta)} - \sqrt{C_1 \log(1/\delta)} \right)^2 \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$

5. Substitute the definition of σ^2 and verify the condition is satisfied:

$$\eta = \frac{\|f(x) - f(x')\|_2^2}{2\sigma^2} = \frac{\varepsilon^2 \|f(x) - f(x')\|_2^2}{2\Delta_2^2 C \log(1/\delta)} \leq \frac{\varepsilon^2}{C_2 \log(1/\delta)}$$



Differential Privacy as a Concentration Property

- Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with privacy loss r.v. $L_{\mathcal{M}, x, x'}$
- Define the cumulant generating function of \mathcal{M} as $\varphi_{\mathcal{M}, x, x'}(s) = \log \mathbb{E}[e^{sL_{\mathcal{M}, x, x'}}]$

Name	Definition	Reference
Concentrated DP (μ, τ) -CDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s\mu + \frac{s^2\tau^2}{2}$	[Dwork and Rothblum, 2016]
Zero-Concentrated DP (ξ, ρ) -zCDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s(\xi + \rho) + s^2\rho$	[Bun and Steinke, 2016]
Rényi DP $(\alpha + 1, \beta)$ -RDP	$x \simeq x'$ $\varphi_{\mathcal{M}, x, x'}(\alpha) \leq \alpha\beta$	[Mironov, 2017]

- Gaussian: For $L \sim \mathcal{N}(\eta, 2\eta)$ the c.g.f. is $\varphi(s) = s\eta + s^2\eta$, i.e. $(0, \eta)$ -zCDP
- Markov: If $\exists s > 0$ such that $\sup_{x \in \mathcal{X}} \varphi_{\mathcal{M}, x, x'}(s) + \log(1/\delta) \leq sc$, then \mathcal{M} is (ϵ, δ) -DP
- Moment accountant: Let $\varphi_i(s)$ be c.g.f. for mechanism \mathcal{M}_i . The mechanism $\mathcal{M}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_k(x))$ has c.g.f. $\varphi_{\mathcal{M}}(s) = \sum_{i=1}^k \varphi_i(s)$ [Abadi et al., 2016]

Differential Privacy as a Concentration Property

- Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with privacy loss r.v. $L_{\mathcal{M}, x, x'}$
- Define the cumulant generating function of \mathcal{M} as $\varphi_{\mathcal{M}, x, x'}(s) = \log \mathbb{E}[e^{sL_{\mathcal{M}, x, x'}}]$

Name	Definition	Reference
Concentrated DP (μ, τ) -CDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s\mu + \frac{s^2\tau^2}{2}$	[Dwork and Rothblum, 2016]
Zero-Concentrated DP (ξ, ρ) -zCDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s(\xi + \rho) + s^2\rho$	[Bun and Steinke, 2016]
Rényi DP $(\alpha + 1, \beta)$ -RDP	$x \simeq x'$ $\varphi_{\mathcal{M}, x, x'}(\alpha) \leq \alpha\beta$	[Mironov, 2017]

- Gaussian: For $L \sim \mathcal{N}(\eta, 2\eta)$ the c.g.f. is $\varphi(s) = s\eta + s^2\eta$, i.e. $(0, \eta)$ -zCDP
- Markov: If $\exists s > 0$ such that $\sup_{x \in \mathcal{X}} \varphi_{\mathcal{M}, x, x'}(s) + \log(1/\delta) \leq sc$, then \mathcal{M} is (ϵ, δ) -DP
- Moment accountant: Let $\varphi_i(s)$ be c.g.f. for mechanism \mathcal{M}_i . The mechanism $\mathcal{M}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_k(x))$ has c.g.f. $\varphi_{\mathcal{M}}(s) = \sum_{i=1}^k \varphi_i(s)$ [Abadi et al., 2016]

Differential Privacy as a Concentration Property

- Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with privacy loss r.v. $L_{\mathcal{M}, x, x'}$
- Define the cumulant generating function of \mathcal{M} as $\varphi_{\mathcal{M}, x, x'}(s) = \log \mathbb{E}[e^{sL_{\mathcal{M}, x, x'}}]$

Name	Definition	Reference
Concentrated DP (μ, τ) -CDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s\mu + \frac{s^2\tau^2}{2}$	[Dwork and Rothblum, 2016]
Zero-Concentrated DP (ξ, ρ) -zCDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s(\xi + \rho) + s^2\rho$	[Bun and Steinke, 2016]
Rényi DP $(\alpha + 1, \beta)$ -RDP	$x \simeq x'$ $\varphi_{\mathcal{M}, x, x'}(\alpha) \leq \alpha\beta$	[Mironov, 2017]

- Gaussian: For $L \sim \mathcal{N}(\eta, 2\eta)$ the c.g.f. is $\varphi(s) = s\eta + s^2\eta$, i.e. $(0, \eta)$ -zCDP
- Markov: If $\exists s > 0$ such that $\sup_{x \simeq x'} \varphi_{\mathcal{M}, x, x'}(s) + \log(1/\delta) \leq s\varepsilon$, then \mathcal{M} is (ε, δ) -DP
- Moment accountant: Let $\varphi_i(s)$ be c.g.f. for mechanism \mathcal{M}_i . The mechanism $\mathcal{M}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_k(x))$ has c.g.f. $\varphi_{\mathcal{M}}(s) = \sum_{i=1}^k \varphi_i(s)$ [Abadi et al., 2016]

Differential Privacy as a Concentration Property

- Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with privacy loss r.v. $L_{\mathcal{M}, x, x'}$
- Define the cumulant generating function of \mathcal{M} as $\varphi_{\mathcal{M}, x, x'}(s) = \log \mathbb{E}[e^{sL_{\mathcal{M}, x, x'}}]$

Name	Definition	Reference
Concentrated DP (μ, τ) -CDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s\mu + \frac{s^2\tau^2}{2}$	[Dwork and Rothblum, 2016]
Zero-Concentrated DP (ξ, ρ) -zCDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s(\xi + \rho) + s^2\rho$	[Bun and Steinke, 2016]
Rényi DP $(\alpha + 1, \beta)$ -RDP	$x \simeq x'$ $\varphi_{\mathcal{M}, x, x'}(\alpha) \leq \alpha\beta$	[Mironov, 2017]

- Gaussian: For $L \sim \mathcal{N}(\eta, 2\eta)$ the c.g.f. is $\varphi(s) = s\eta + s^2\eta$, i.e. $(0, \eta)$ -zCDP
- Markov: If $\exists s > 0$ such that $\sup_{x \simeq x'} \varphi_{\mathcal{M}, x, x'}(s) + \log(1/\delta) \leq s\varepsilon$, then \mathcal{M} is (ε, δ) -DP
- Moment accountant: Let $\varphi_i(s)$ be c.g.f. for mechanism \mathcal{M}_i . The mechanism $\mathcal{M}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_k(x))$ has c.g.f. $\varphi_{\mathcal{M}}(s) = \sum_{i=1}^k \varphi_i(s)$ [Abadi et al., 2016]

Differential Privacy as a Concentration Property

- Let $\mathcal{M} : X \rightarrow Y$ be a randomized mechanism with privacy loss r.v. $L_{\mathcal{M}, x, x'}$
- Define the cumulant generating function of \mathcal{M} as $\varphi_{\mathcal{M}, x, x'}(s) = \log \mathbb{E}[e^{sL_{\mathcal{M}, x, x'}}]$

Name	Definition	Reference
Concentrated DP (μ, τ) -CDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s\mu + \frac{s^2\tau^2}{2}$	[Dwork and Rothblum, 2016]
Zero-Concentrated DP (ξ, ρ) -zCDP	$x \simeq x'$, $s > 0$ $\varphi_{\mathcal{M}, x, x'}(s) \leq s(\xi + \rho) + s^2\rho$	[Bun and Steinke, 2016]
Rényi DP $(\alpha + 1, \beta)$ -RDP	$x \simeq x'$ $\varphi_{\mathcal{M}, x, x'}(\alpha) \leq \alpha\beta$	[Mironov, 2017]

- Gaussian: For $L \sim \mathcal{N}(\eta, 2\eta)$ the c.g.f. is $\varphi(s) = s\eta + s^2\eta$, i.e. $(0, \eta)$ -zCDP
- Markov: If $\exists s > 0$ such that $\sup_{x \simeq x'} \varphi_{\mathcal{M}, x, x'}(s) + \log(1/\delta) \leq s\varepsilon$, then \mathcal{M} is (ε, δ) -DP
- Moment accountant: Let $\varphi_i(s)$ be c.g.f. for mechanism \mathcal{M}_i . The mechanism $\mathcal{M}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_k(x))$ has c.g.f. $\varphi_{\mathcal{M}}(s) = \sum_{i=1}^k \varphi_i(s)$ [Abadi et al., 2016]

Differential Privacy Without a Trusted Curator

Issues with the Trusted Curator Assumption

- ▶ *Single point of failure*: a DP curator might have other security vulnerabilities
- ▶ *Conflicting incentives*: valuable the data provides incentives for the curator to misbehave
- ▶ *Requires agreement*: a large number of individuals need to agree on who to trust

Randomized response: recall in $(y_1, \dots, y_n) = RR_\epsilon(x_1, \dots, x_n)$ each y_i depends only on x_i

Multi-Party and Local Differential Privacy

- ▶ Dataset x distributed among m parties, party i owns \vec{x}_i
- ▶ Analyst initiates randomized protocol $\Pi : X \rightarrow Y$ that interacts with the parties
- ▶ All the outputs produced by party i during $\Pi(x)$ determine a mechanism $\mathcal{M}_i(\vec{x}_i)$
- ▶ Π is *multi-party* (ϵ, δ) -DP if each \mathcal{M}_i is (ϵ, δ) -DP
- ▶ When each \vec{x}_i has size one we talk about *local DP*
- ▶ Utility loss: the difference between $\mathcal{O}(1/n)$ (Laplace) and $\mathcal{O}(1/\sqrt{n})$ (RR) is characteristic of local DP



Differential Privacy Without a Trusted Curator

Issues with the Trusted Curator Assumption

- ▶ *Single point of failure*: a DP curator might have other security vulnerabilities
- ▶ *Conflicting incentives*: valuable the data provides incentives for the curator to misbehave
- ▶ *Requires agreement*: a large number of individuals need to agree on who to trust

Randomized response: recall in $(y_1, \dots, y_n) = RR_\epsilon(x_1, \dots, x_n)$ each y_i depends only on x_i

Multi-Party and Local Differential Privacy

- ▶ Dataset x distributed among m parties, party i owns \vec{x}_i
- ▶ Analyst initiates randomized protocol $\Pi : X \rightarrow Y$ that interacts with the parties
- ▶ All the outputs produced by party i during $\Pi(x)$ determine a mechanism $\mathcal{M}_i(\vec{x}_i)$
- ▶ Π is *multi-party* (ϵ, δ) -DP if each \mathcal{M}_i is (ϵ, δ) -DP
- ▶ When each \vec{x}_i has size one we talk about *local DP*
- ▶ Utility loss: the difference between $\mathcal{O}(1/n)$ (Laplace) and $\mathcal{O}(1/\sqrt{n})$ (RR) is characteristic of local DP



Differential Privacy Without a Trusted Curator

Issues with the Trusted Curator Assumption

- ▶ *Single point of failure*: a DP curator might have other security vulnerabilities
- ▶ *Conflicting incentives*: valuable the data provides incentives for the curator to misbehave
- ▶ *Requires agreement*: a large number of individuals need to agree on who to trust

Randomized response: recall in $(y_1, \dots, y_n) = RR_\varepsilon(x_1, \dots, x_n)$ each y_i depends only on x_i

Multi-Party and Local Differential Privacy

- ▶ Dataset x distributed among m parties, party i owns \vec{x}_i
- ▶ Analyst initiates randomized protocol $\Pi : X \rightarrow Y$ that interacts with the parties
- ▶ All the outputs produced by party i during $\Pi(x)$ determine a mechanism $\mathcal{M}_i(\vec{x}_i)$
- ▶ Π is *multi-party* (ε, δ) -DP if each \mathcal{M}_i is (ε, δ) -DP
- ▶ When each \vec{x}_i has size one we talk about *local DP*
- ▶ **Utility loss**: the difference between $\mathcal{O}(1/n)$ (Laplace) and $\mathcal{O}(1/\sqrt{n})$ (RR) is characteristic of local DP

Outline

1. We Need Mathematics to Study Privacy? Seriously?
2. Differential Privacy: Definition, Properties and Basic Mechanisms
3. Differentially Private Machine Learning: ERM and Bayesian Learning
4. Variations on Differential Privacy: Concentrated DP and Local DP
5. Final Remarks

Beyond This Tutorial...

Additional Results

- ▶ More basic mechanisms: sparse vector technique and other selection mechanisms, private data structures
- ▶ General theorems: everything is randomized response, lower bounds on utility, computational hardness, optimal mechanisms, connections to generalization
- ▶ Database perspective: answering multiple queries on the same data, adaptive vs. non-adaptive queries
- ▶ When global sensitivity is atypical: smoothed sensitivity, randomized DP
- ▶ Other privacy definitions: location privacy, pan DP, pufferfish privacy

Suggested Readings

- ▶ “The Algorithmic Foundations of Differential Privacy” [Dwork and Roth, 2014]
- ▶ “The Complexity of Differential Privacy” [Vadhan, 2017]

Some Open Research Directions

Bounds vs. Algorithms

- ▶ Few privacy analysis are tight: randomized response, Laplace mechanism, ϵ -DP exponential mechanism
- ▶ Most complex mechanisms add too much noise (constants in bounds matter!)
- ▶ Alternative: calibrate noise using “exact” numerical computations instead of bounds
- ▶ Challenges: concentration bounds vs. exact densities, compositions, sub-sampling and other mixtures, approximate sampling

Correctness and Attacks

- ▶ Given a mechanism, it is not possible to test empirically if it is DP
- ▶ We can only resort to mathematical proofs to establish correctness (can be automated?)
- ▶ But we should have sanity-check to tools to break DP of candidate implementations
- ▶ Challenge: from pseudo-code to implementation things can go wrong (floating-point $\neq \mathbb{R}$)

Conclusion

- ▶ Differential privacy provides a formal notion of privacy satisfying many desirable properties
 - ▶ Precise quantification of the privacy-utility trade-off
 - ▶ Robustness against powerful adversaries (eg. in the presence of side knowledge)
 - ▶ Applicable to a wide range of data analysis problems
- ▶ Mature research field with a rich toolbox of mechanism design strategies
- ▶ Natural starting point for application-specific privacy guarantees
- ▶ Several real-world deployments and open source tools
 - ▶ Google Chrome's RAPPOR
 - ▶ Apple's iOS 10
 - ▶ U.S. Census Bureau
 - ▶ GUPT, Microsoft's PINQ, Uber's FLEX

References I

-  Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy.
In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM.
-  Bassily, R., Smith, A. D., and Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds.
In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 464–473.
-  Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds.
In *Theory of Cryptography Conference*, pages 635–658. Springer.
-  Dwork, C. (2006). Differential privacy.
In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 1–12.

References II



Dwork, C., McSherry, F., Nissim, K., and Smith, A. D. (2006).

Calibrating noise to sensitivity in private data analysis.

In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 265–284.



Dwork, C. and Roth, A. (2014).

The algorithmic foundations of differential privacy.

Foundations and Trends® in Theoretical Computer Science, 9(3–4):211–407.



Dwork, C. and Rothblum, G. N. (2016).

Concentrated differential privacy.

arXiv preprint arXiv:1603.01887.



Foulds, J. R., Geumlek, J., Welling, M., and Chaudhuri, K. (2016).

On the theory and practice of privacy-preserving bayesian data analysis.

In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI 2016, June 25-29, 2016, New York City, NY, USA*.

References III

-  Jain, P. and Thakurta, A. G. (2014).
(near) dimension independent risk bounds for differentially private learning.
In *International Conference on Machine Learning*, pages 476–484.
-  McSherry, F. and Talwar, K. (2007).
Mechanism design via differential privacy.
In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103.
IEEE.
-  Minami, K., Arai, H., Sato, I., and Nakagawa, H. (2016).
Differential privacy without sensitivity.
In *Advances in Neural Information Processing Systems*, pages 956–964.
-  Mironov, I. (2017).
Renyi differential privacy.
arXiv preprint arXiv:1702.07476.

References IV

-  Park, M., Foulds, J. R., Chaudhuri, K., and Welling, M. (2016).
Variational bayes in private settings (VIPS).
CoRR, abs/1611.00340.
-  Talwar, K., Thakurta, A., and Zhang, L. (2014).
Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry.
CoRR, abs/1411.5417.
-  Vadhan, S. P. (2017).
The complexity of differential privacy.
In *Tutorials on the Foundations of Cryptography.*, pages 347–450.
-  Wang, Y., Fienberg, S. E., and Smola, A. J. (2015).
Privacy for free: Posterior sampling and stochastic gradient monte carlo.
In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2493–2502.

References V



Warner, S. L. (1965).

Randomized response: A survey technique for eliminating evasive answer bias.

Journal of the American Statistical Association, 60(309):63–69.



Wu, X., Kumar, A., Chaudhuri, K., Jha, S., and Naughton, J. F. (2016).

Differentially private stochastic gradient descent for in-rdbms analytics.

CoRR, abs/1606.04722.



Zhang, Z., Rubinstein, B. I. P., and Dimitrakakis, C. (2016).

On the differential privacy of bayesian inference.

In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2365–2371.

A Short Tutorial on Differential Privacy

Borja Balle

Amazon Research Cambridge

The Alan Turing Institute — January 26, 2018

Notes on Differential Privacy

Advanced Database Management, UNIBG

Stefano Paraboschi

Current significant limitations of DP

- (1) the necessity to keep a detailed track of the privacy budget and adapt the noise level based on the remaining budget
- (2) the huge difference in the level of noise between the local and centralized models
- (3) the limited scope of the query result that is compatible with adequate noise
- (4) the choice of values for privacy parameter epsilon (and delta, in the (epsilon,delta) model).

Advanced Databases

7

Data Warehouses

An Environment for Data Analysis

- **DATA WAREHOUSE**

- A structured description of all those data that are necessary for a strategic analysis of the trends and the behaviour of a firm

- **ON-LINE ANALYTICAL PROCESSING (OLAP)**

- The name given to analysis activities (it is contrasted to On Line Transaction Processing, OLTP)

Data analysis

OLTP vs OLAP

The problem

- A promise of relational technology “flexible data access”:
 - A tool for the end user
 - to express all types of query
- Relational technology did not provide:
 - Complexity and rigidity of applications
 - Emphasis to OLTP
- Consequences:
 - Data volumes for operational procedures
 - Limited us of data for strategy

Functional separation

- Operations environments:
«on line» data management, to support update transactions

On Line Transaction Processing (OLTP)

- Analysis environment:
«off line» data management, dedicated to statistical queries and analyses

On Line Analytical Processing (OLAP)

OLTP: On Line Transaction Processing

- Traditional transaction processing, which automatizes the operational procedures of the organization
 - Predefined and relatively simple operations
 - Each operation involves «a few» data items
 - Detailed data, up to date
 - «Acid» transactional properties (atomicity, consistency, isolation, durability) are essential

OLAP: On Line Analytical Processing

- Data processing for decision support
 - Complex and random (unpredictable) operations
 - Each operation can involve multiple data
 - Aggregated, historical, and potentially not up-to-date data
 - «Acid» properties are not relevant, because operations are read-only

OLTP vs OLAP

- Configuration of a system dedicated to one of them is manageable
- It is extremely difficult to manage both workloads on the same system
- A variety of reasons:
 - Dishomogeneous users and requirements
 - Technical aspects

OLTP e OLAP

	OLTP	OLAP
User	clerk	manager
Function	Daily operations	Decision support
Design	Application oriented	Data oriented
Data	current, updated, detailed, relational, homogeneous	historical, aggregate, multidimensional, heterogeneous
Use	repetitive	casual
Access	read-write, indexed	read, sequential
Unit of work	Short transaction	Complex query
#Acc. records	tens	millions
#users	>thousand	tens/hundreds
Size	100MB – 1GB	>1TB
Metrics	tps	qph/response time

Technical reasons of the OLTP/OLAP conflict

- Lock conflict
 - OLTP: many fast transactions with exclusive locks
 - OLAP: a few long transactions with shared locks
 - OLTP+OLAP
 - either the OLTP transactions are delayed
 - or OLAP queries cannot be executed

Technical reasons of the OLTP/OLAP conflict

- Index use
 - OLTP: a few and only when clearly needed
 - OLAP: many, to deal with any possible use
 - OLTP+OLAP
 - Either OLTP transactions are slowed down by the need to update many indexes
 - or OLAP queries do not have available the indexes they need

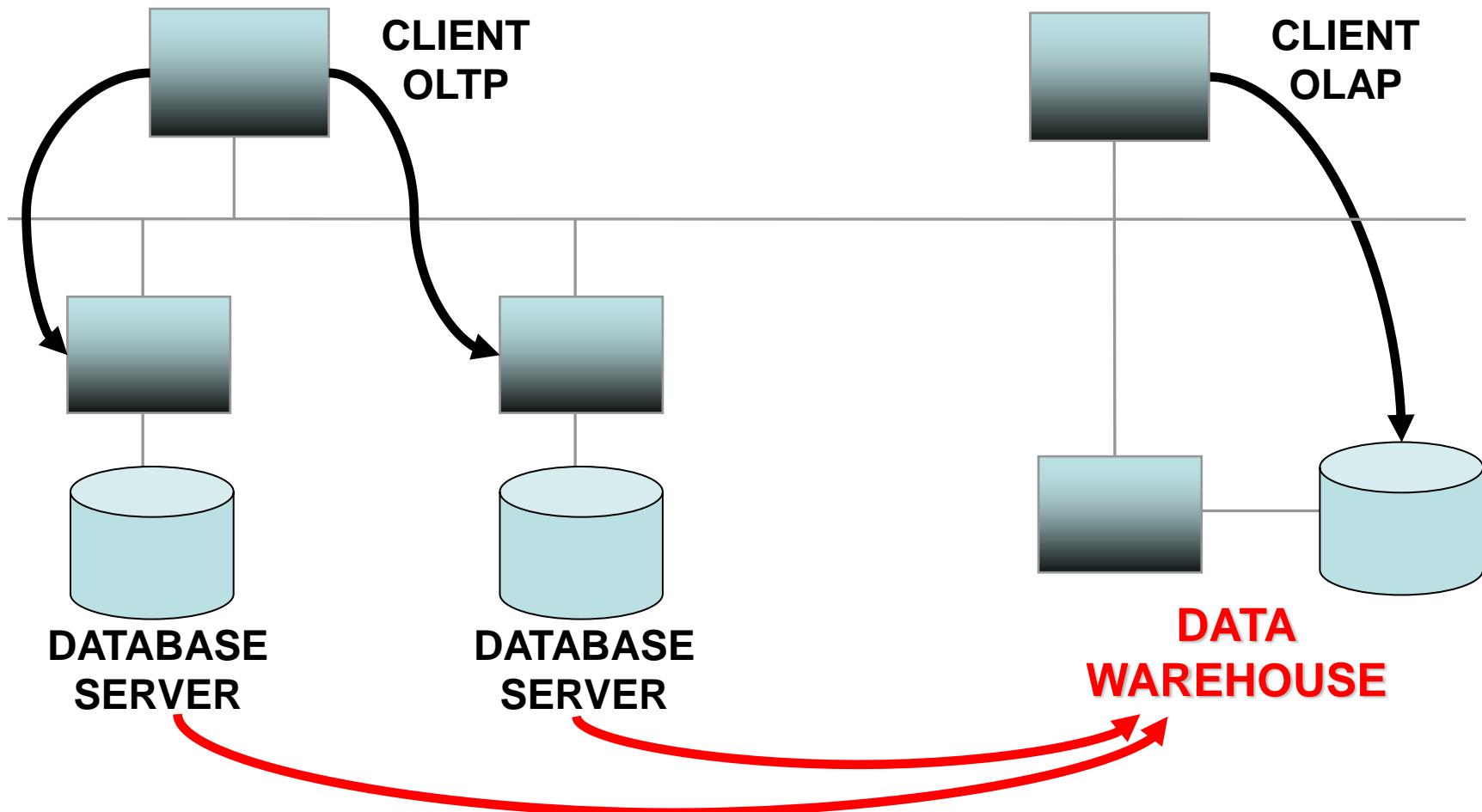
Technical reasons of the OLTP/OLAP conflict

- Query precomputations (i.e., materialized views)
 - OLTP: rarely, for consistency and load requirements
 - OLAP: key aspect to improve query response times
- Differences in the logical model
 - OLTP: high fragmentation and a large number of normalized tables
 - OLAP: few denormalized tables
- Different join algorithms

Considerations on the OLTP/OLAP conflict

- The conflict is intrinsic
- It does not disappear with the increase in computational power, rather it can increase
- The best solution consists in the separation of the two environments
 - This leads to the creation of Data Warehouses
- Asynchronicity of updates is fundamental

Interaction between OLTP and OLAP



Data Warehouse

Analysis environment: Data Warehouse

- **DATA WAREHOUSE:**
organized description of all the data required for
a strategic analysis of the organization behavior
- Techniques: multidimensional analysis,
data mining

Data warehouse

- A data base
 - Mainly used to support management decisions
 - integrated — organization-level, not department-level
 - data-oriented — not application-oriented
 - historical — with a wide temporal coverage and (usually) attributes on the time dimension
 - not volatile — data are loaded and updated off-line
 - separate from operational data bases

... integrated ...

- The data of interest originate from all possible data sources — each data item can come from one or more of them
- The data warehouse represents data in a univocal way — reconciling the heterogeneity of the distinct representations
 - names
 - codes
 - multiple representations

... data-oriented ...

- Operational data bases are built to support specific operational processes and applications
 - production
 - sales
- The data warehouse is built around the major entities of the organization information resources
 - product
 - client

... historical data ...

- Operational data bases keep the current value of information
- Temporal horizon is in the order of a few months
- In data warehouses it is of interest to analyze the historical evolution of the data
- Temporal horizon is in the order of years

... not volatile ...

- In an operational data base, data are
 - Accessed, inserted, modified, deleted
 - a few records at a time
- In a data warehouse, we have
 - «daily» access operations and queries
 - «nightly» load and update operations
- which are applied over millions of records

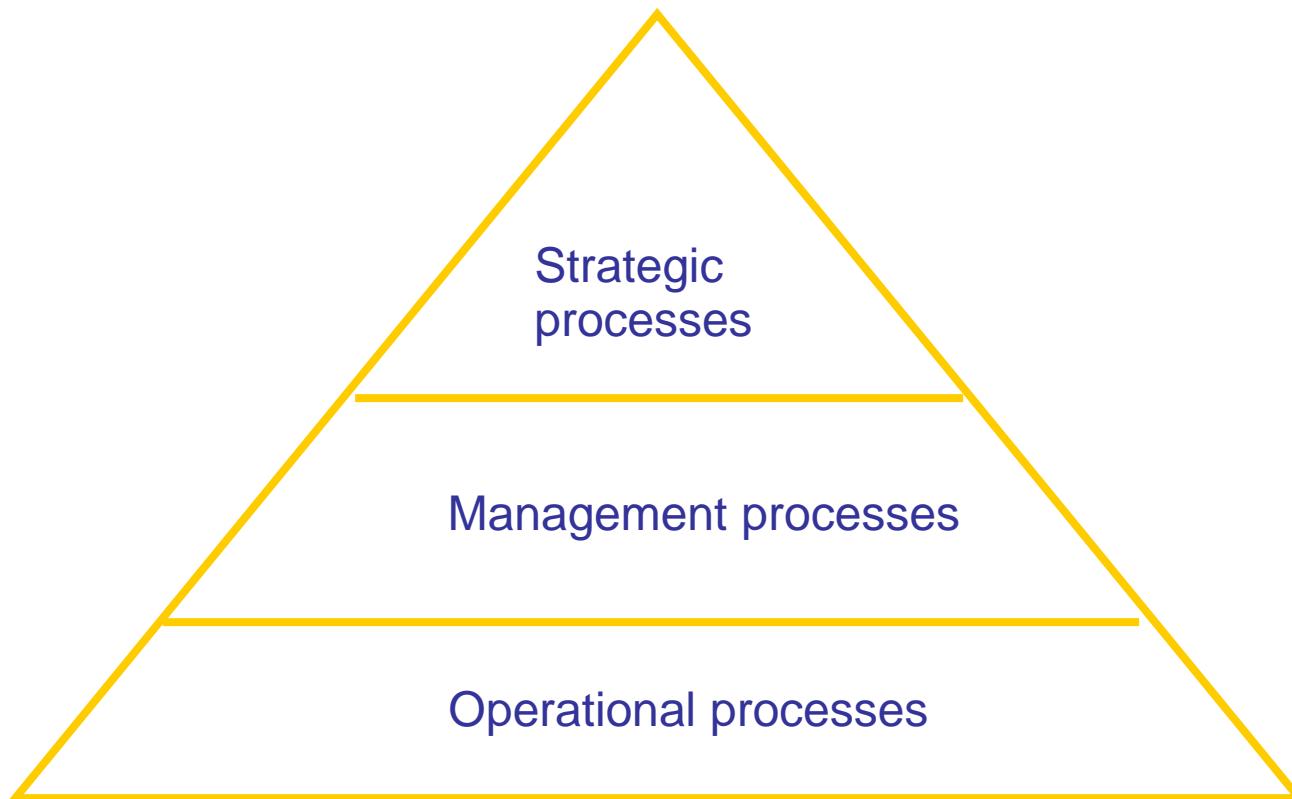
... a separate data base ...

- Due to several reasons
 - technical reasons we saw
 - There is no single operational database with all the data needed for analysis
 - The data base must be integrated
 - The data are in any case different
 - Historical data must be maintained
 - Aggregate data must be stored
 - Data analysis requires special organizations and access methods
 - General negative performance impact if there is no separation

Critical success factors

- Data replication must not have a great impact on the transactional system
- Data must be loaded in the assigned time window
- The solution must be scalable
- It must be accepted by users (presentation)
- Replicated data must be correct
- Standards should be used
- The data model should be consistent with reality

Processes

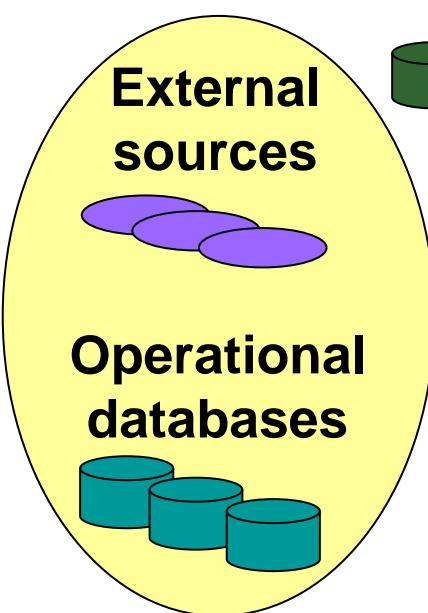


Management control systems

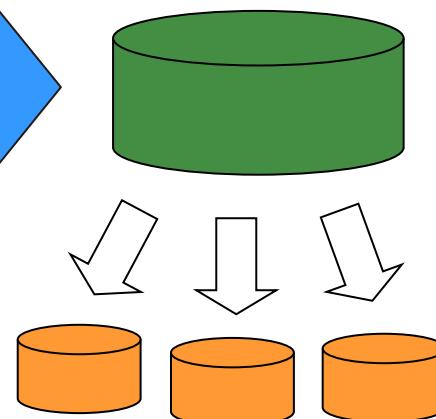
- MCSs share some features with OLAP applications
 - Mainly read-only
 - Need to compute aggregates
 - Comparison with historical series
- But they are different!
 - Static queries (not ad-hoc queries, rather fixed reports)
 - Need to keep the detail of the schema
- They can share information, but they must be distinct projects

An Architecture for Data Warehousing

Monitoring & Administration



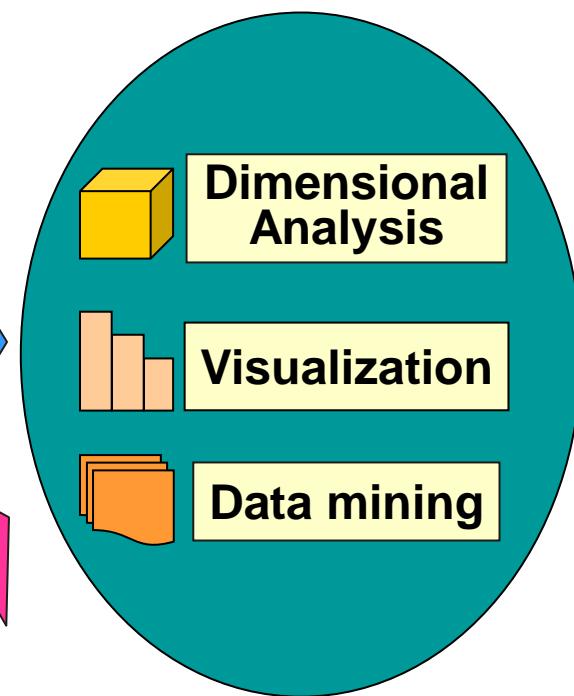
Enterprise Data Warehouse



Data sources

Data Mart

Analysis tools



DW and Data Marts

- A DW often integrated several Data Marts
- Users typically access a specific Data Mart
- Data Marts share data among themselves
- Each Data Mart is responsible of a specific aspect of the organization reality

Multidimensionale Model

Star model

- The ***star model*** is used for each Data Mart
 - Also known as ***multi-dimensional schema***
- It is a conceptual model which introduces some restrictions
- Advantages:
 - Availability of suitable specific query interfaces
 - Good performance
 - Straightforward design of the relational schema

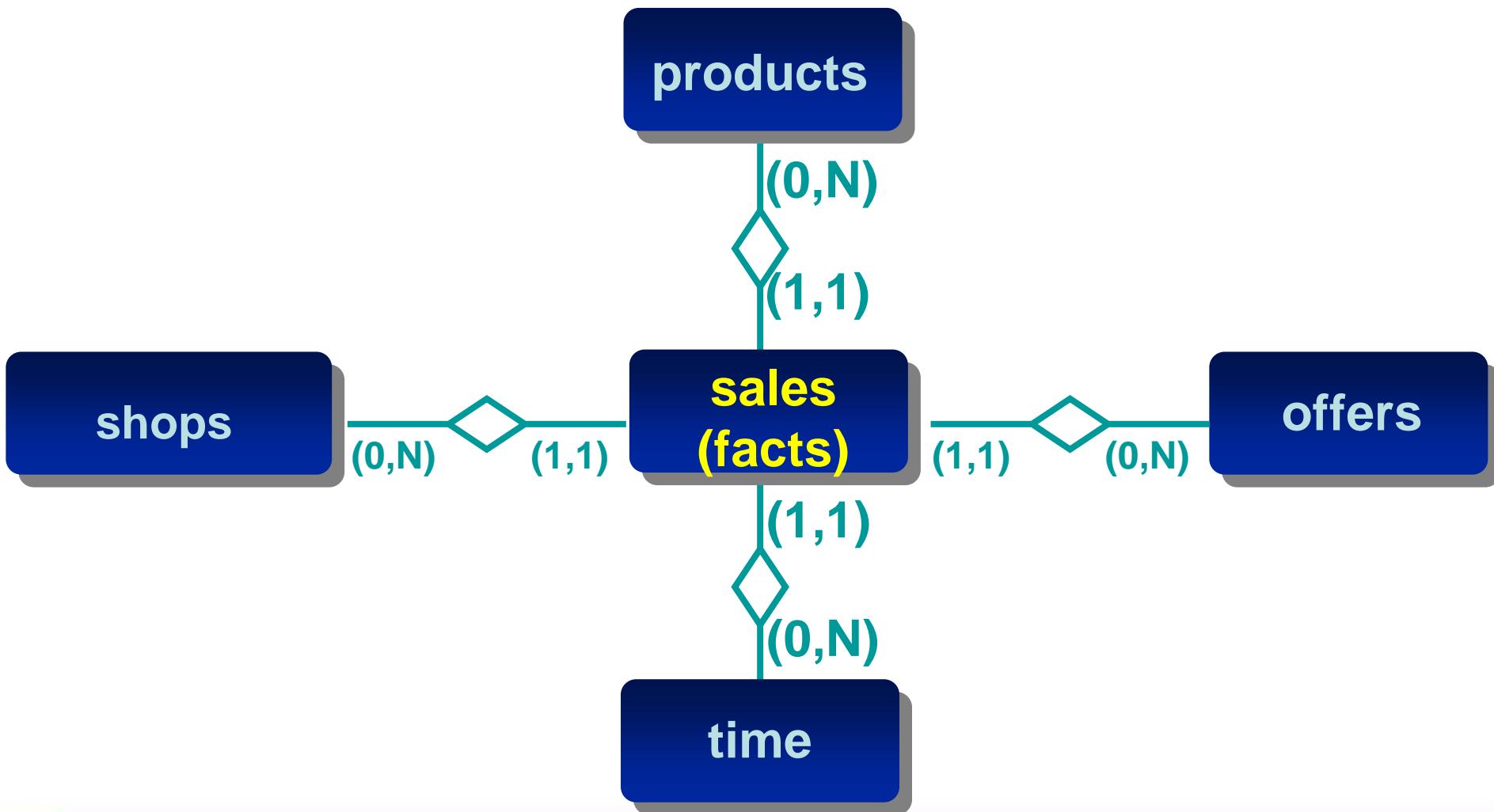
Multi-dimensional representation

- Relevant concepts:
 - **fact** — an aspect which is crucial for the analysis
 - **measure** — an atomic property of a fact
 - **dimension** — a specific perspective for the analysis

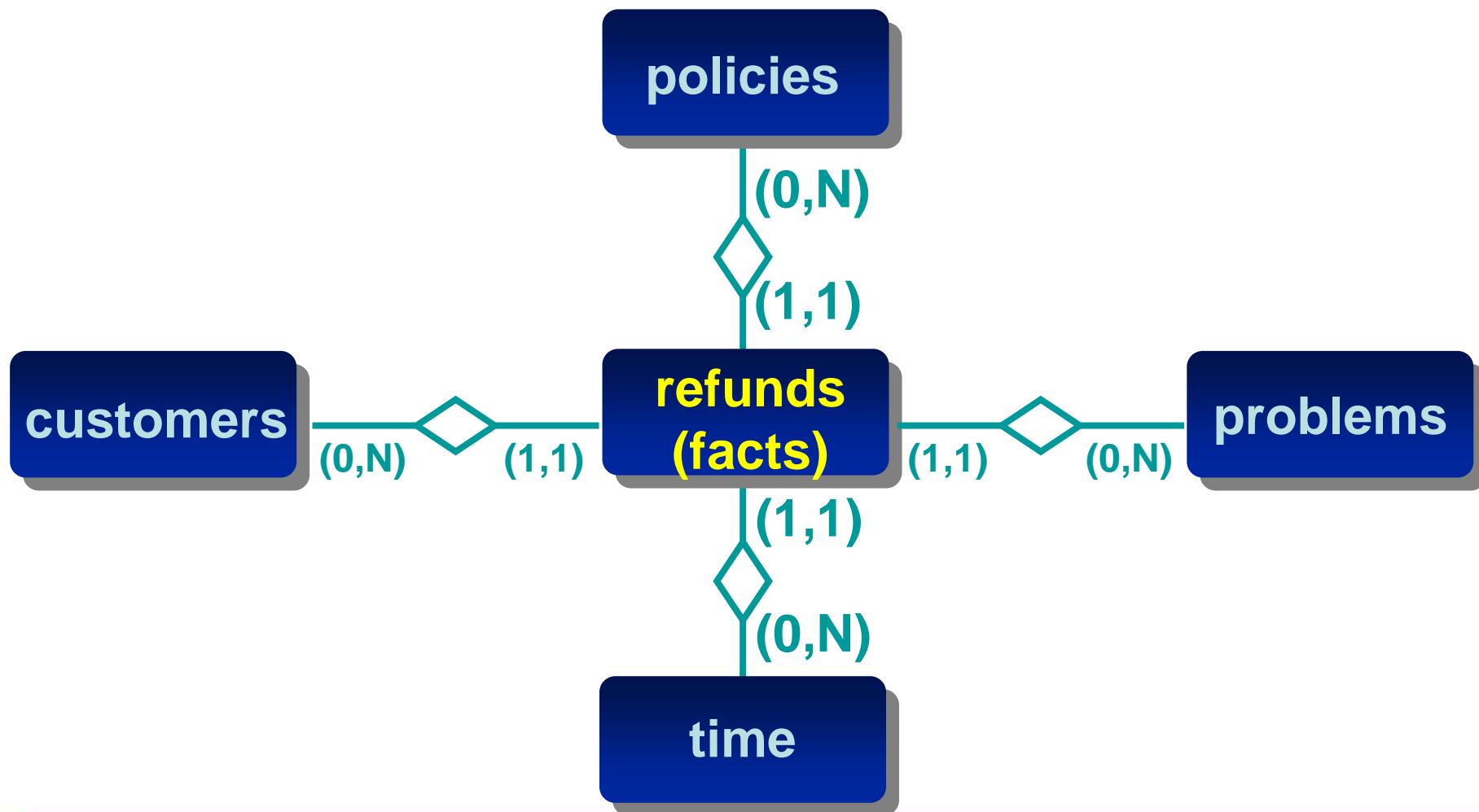
Examples of facts/measures/dimensions

- Retail shops:
 - Sales
 - Quantity, price
 - Product, time, zone
- Telephone service:
 - Phone call
 - Cost, duration
 - Caller, answerer, time

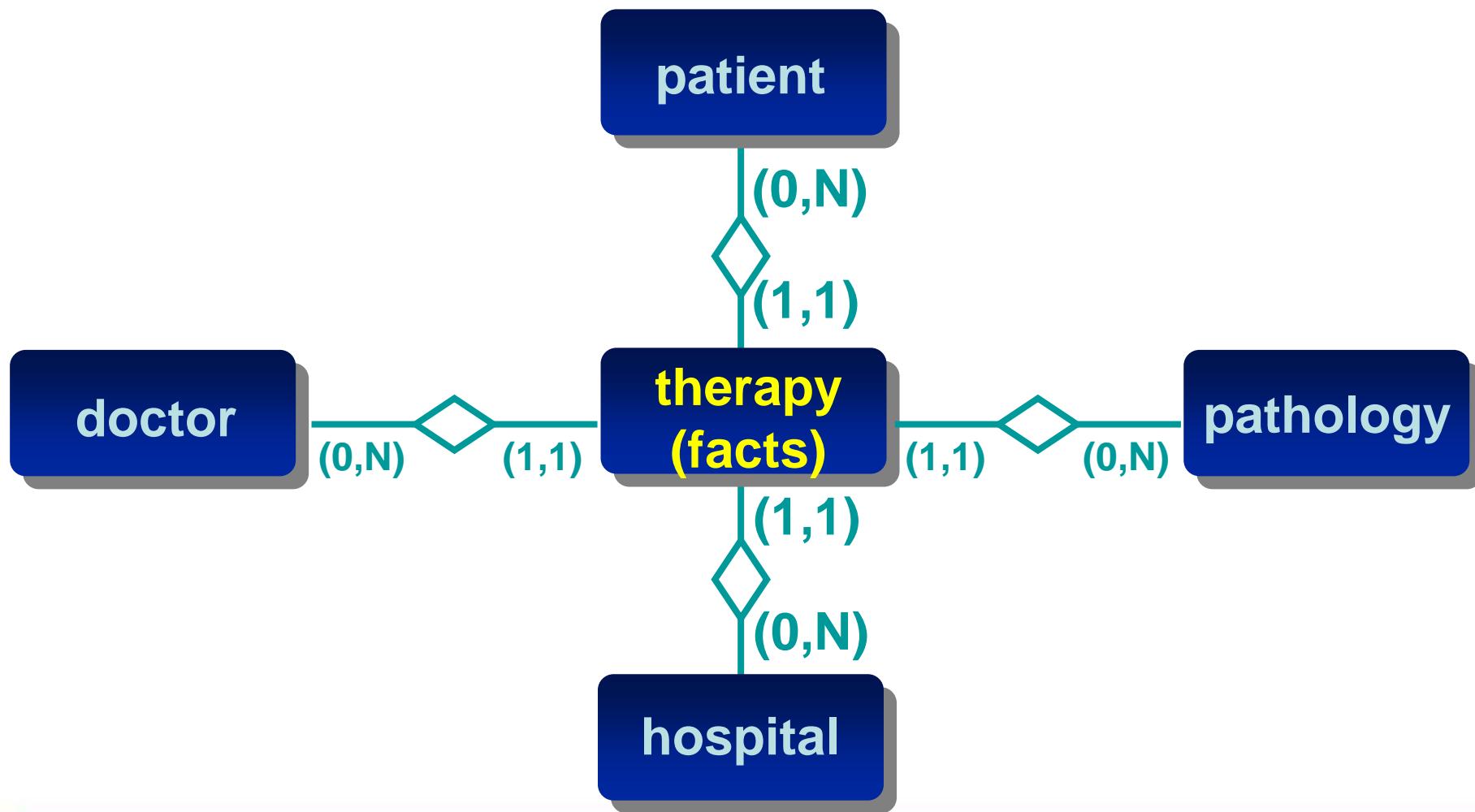
An example: sales management



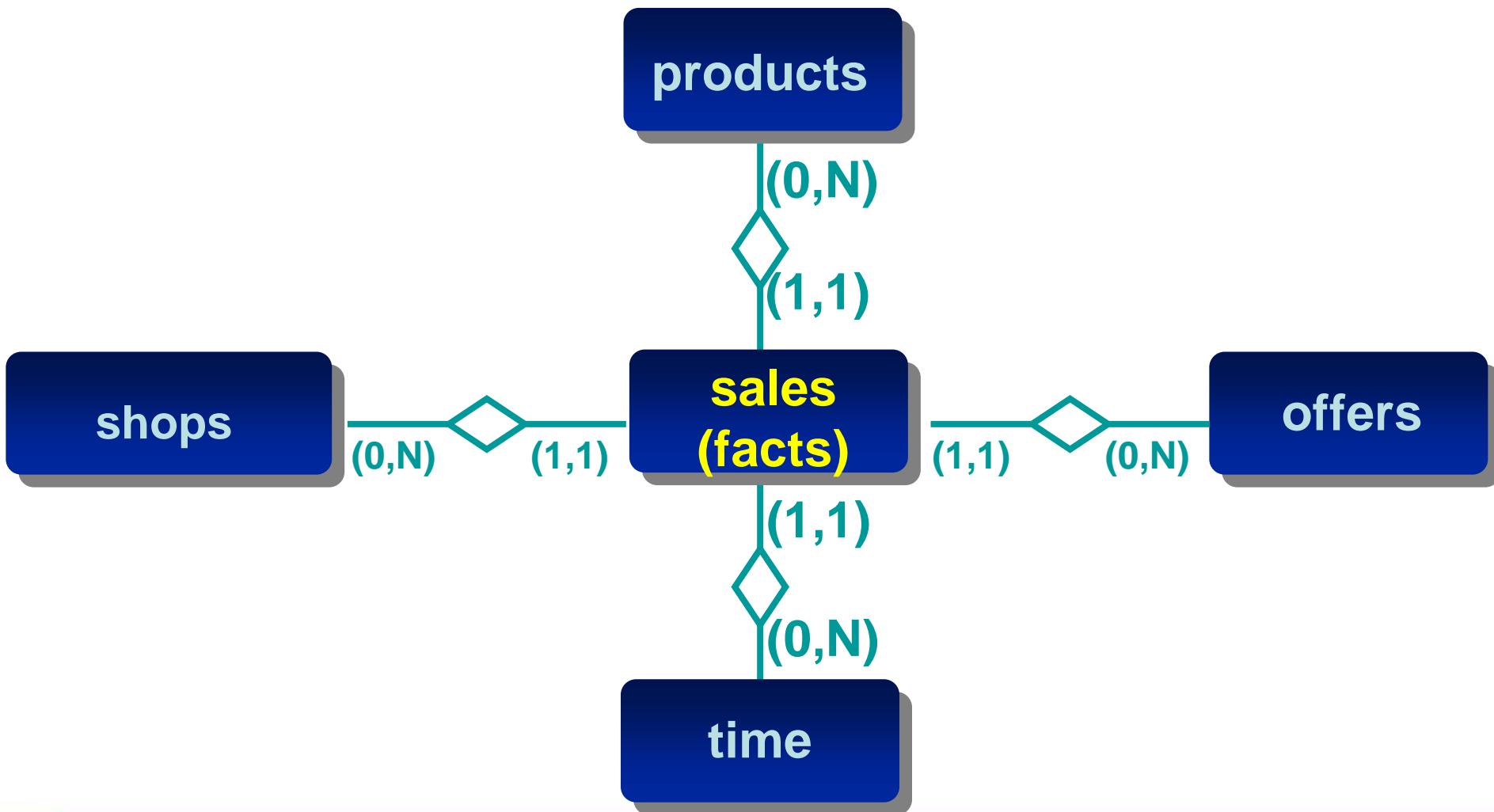
Another example: reimbursements



Another example: therapies



Consider again the sales management schema



Facts: sales

Product-ID

Shop-ID

Time-ID

Offer-ID

Total-proceeds

Quantity

Unit-proceeds

First dimension: products

Product-ID

Category

Sub-Category

Brand

Packing

Weight

Size

Provider

Second dimension: shops

Shop-ID

Name

Address

City

Sales-District

Phone

Manager-Name

Size

Logistics

Third dimension: time

Time-ID

Day-in-Week

Day-in-Month

Day-in-Year

Week-in-Month

Week-in-Year

Month-in-Year

Season

Flag-WorkingDay

Flag-Sunday

Fourth dimension: offers

Offer-ID

Offer-name

Discount-Type

Discount-Percentage

Advertisement

Flag-Coupon

Start-Date

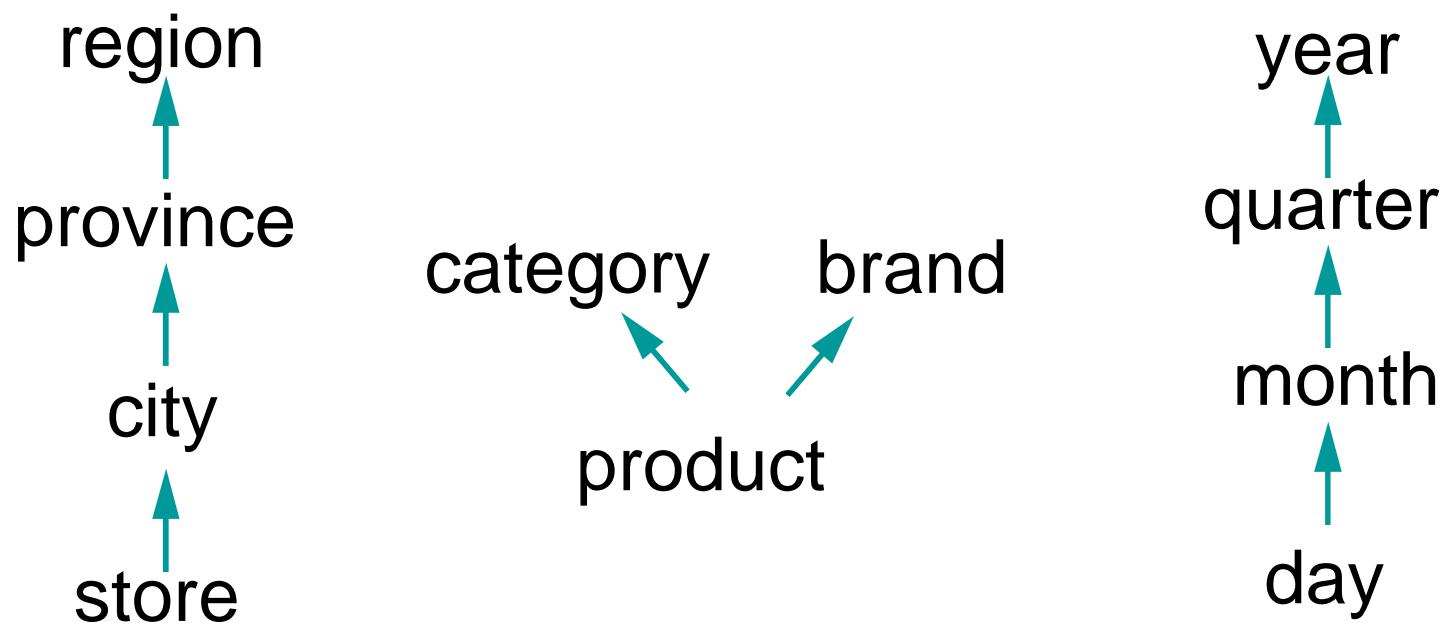
End-Date

Cost

Agency

Dimensions and hierarchies

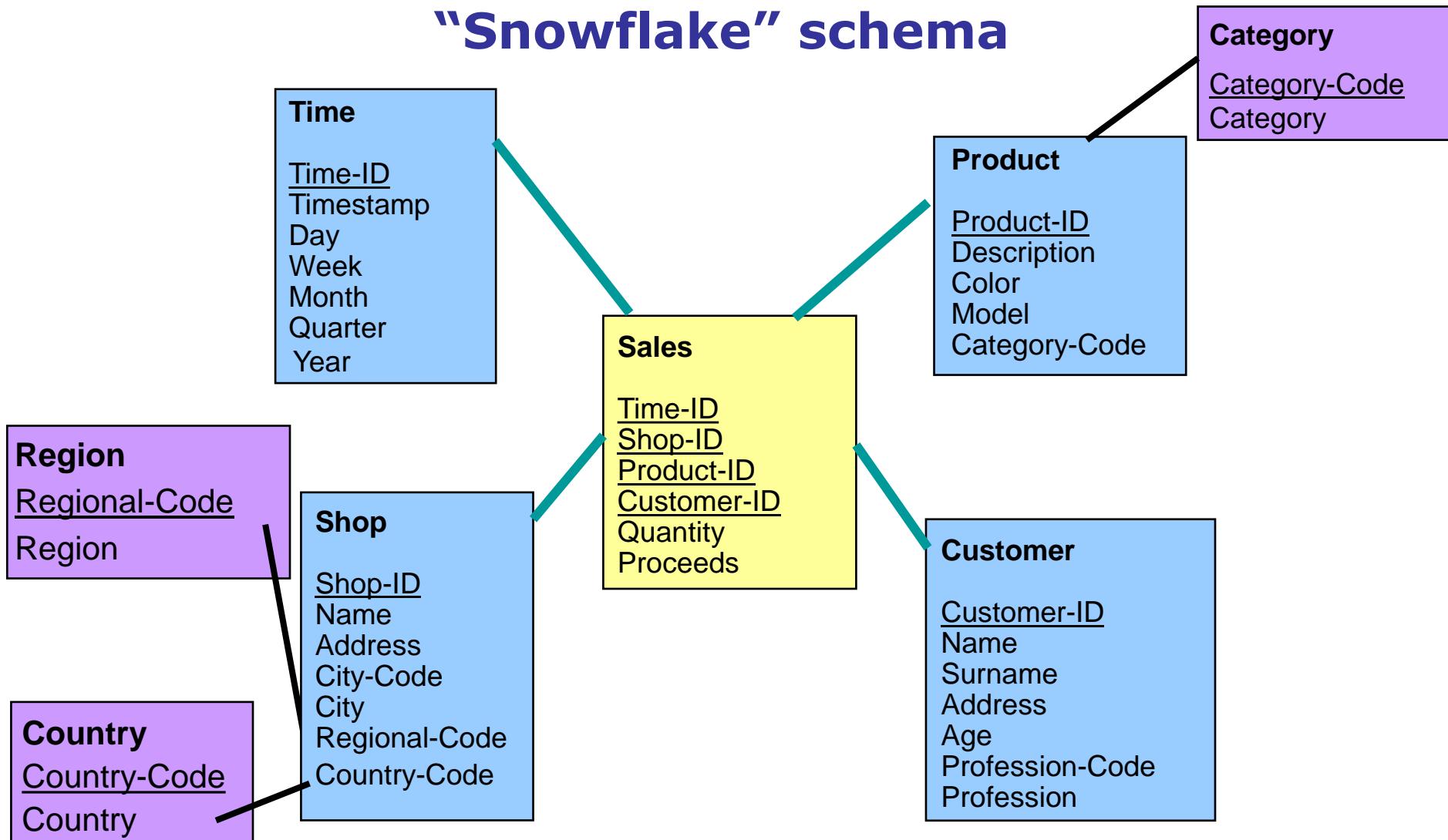
- Each dimension is organized in a hierarchy, which represent possible aggregation levels



Snow-flake model

- It extends the star model
- It permits to reduce excessive redundancy in dimensions
- Starting from the fact table, it is always possible to reach all the dimension (sub)-tables, always passing through n:1 relationships

“Snowflake” schema



Multi-dimensional data representation

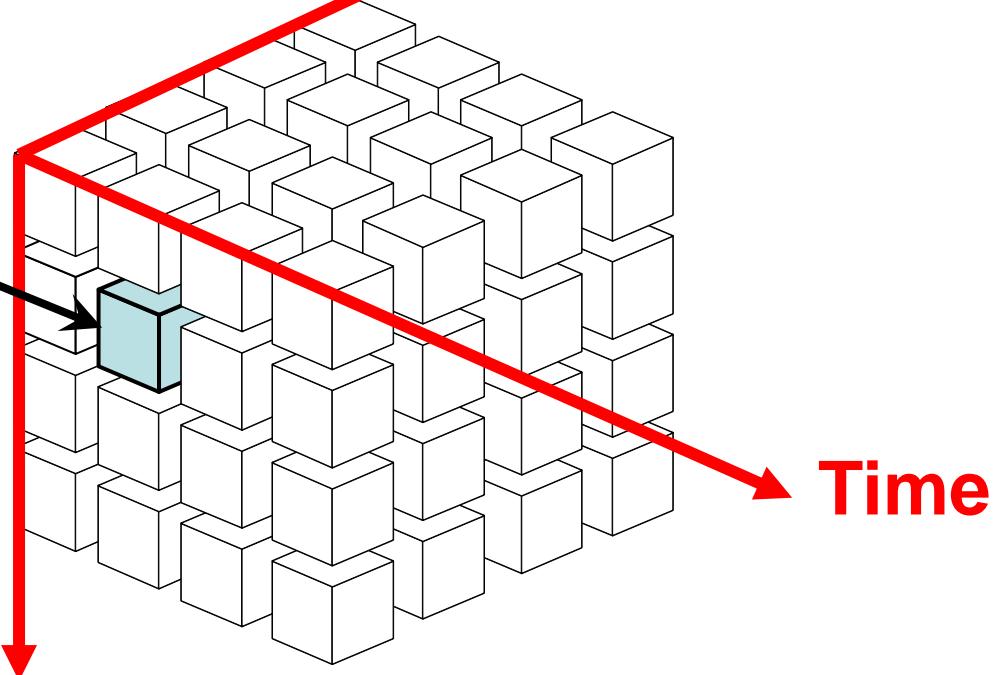
SALES

Quantity

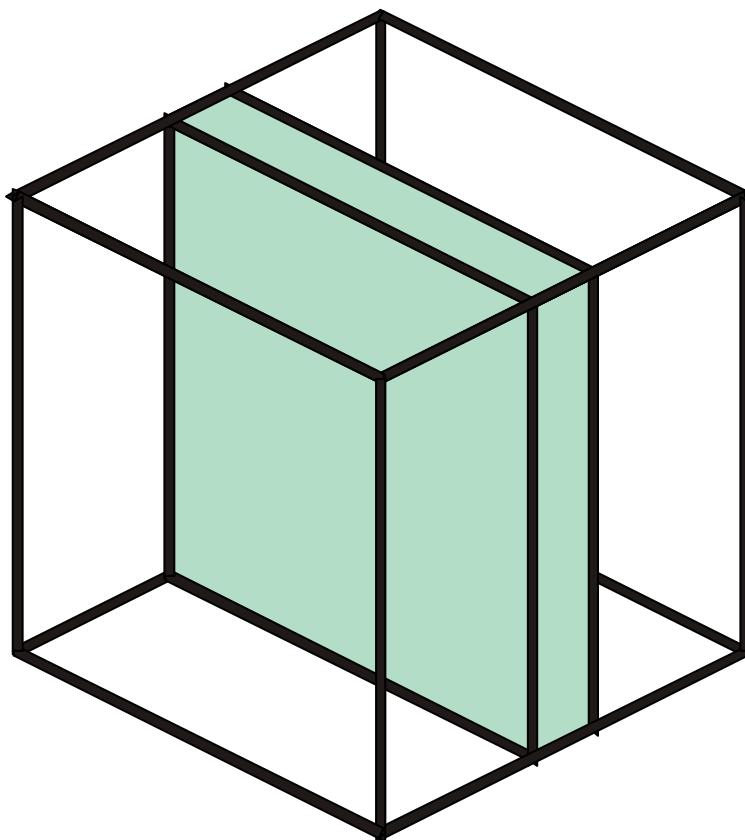
Products

Shops

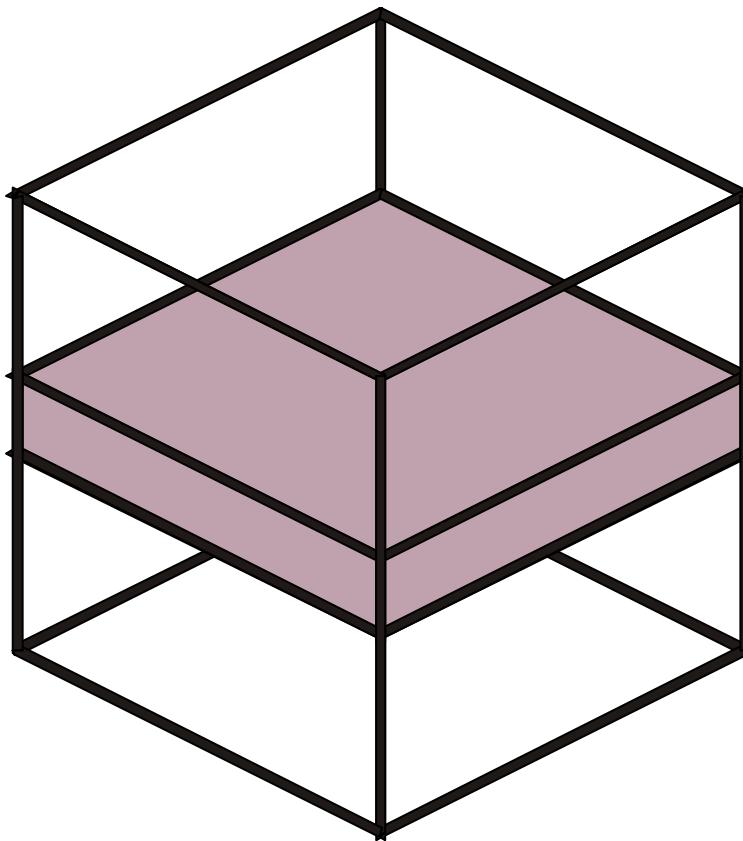
Time



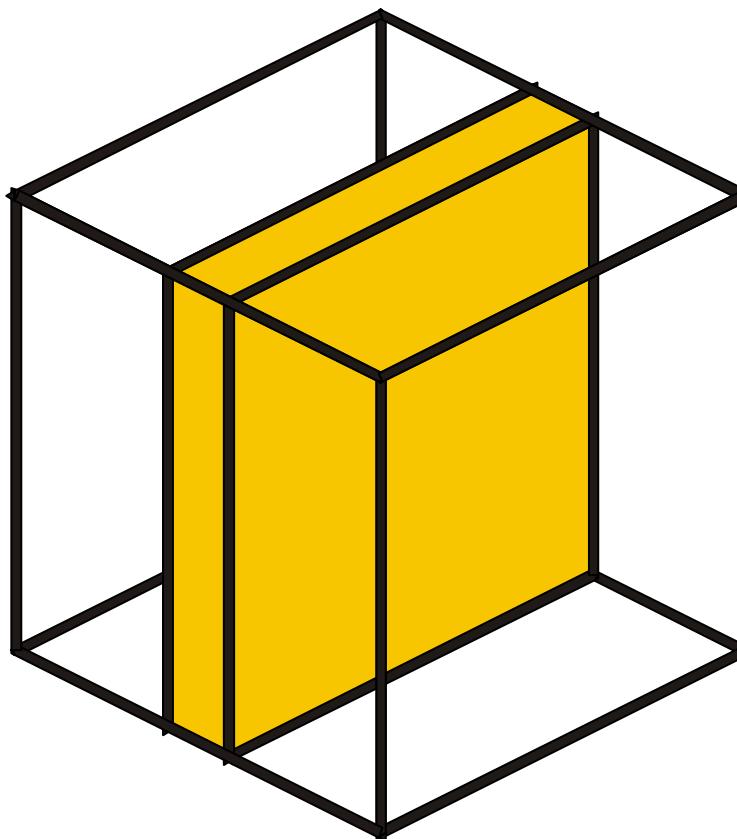
- The regional manager studies the sales of all products in all periods w.r.t. the shops of his region



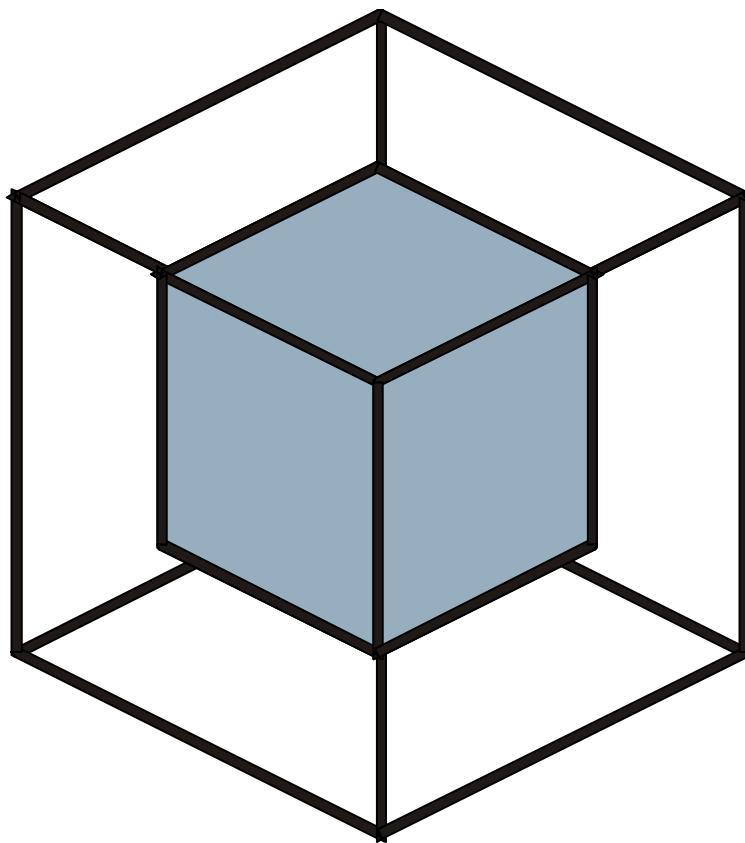
- The product manager studies the sales of a product in all periods and in all shops



- The financial manager studies the sales of all products in all shops, comparing the current period with the previous one



- The strategic manager focuses on one category of products, one area and a limited period of time



Data Visualization

- Data are visualized and rendered graphically, so as to be easy to understand
- Common means of visualization:
 - Tables
 - Pie/doughnut charts
 - Column bar/histograms
 - Line charts
 - 3D surfaces
 - Bubble charts
 - Area blocks
 - Cylinders/cones/pyramids
 - ...

Example of query with a browser

Offer	period	zone	product	dimension of analysis
Pay 2 & buy 3	March	north	milk	total-proceeds
40% discount	April	east	bread	quantity
20% discount	May	west	pasta	unit-proceeds
1-free-mug (...)	
.....				
	February/ April		pasta	sum(proceeds) sum(quantity)

The “same” query in SQL

```
select c1, c2, aggr(c3), aggr(c4)
from facts, dim1, dim2
where join-predicate(facts, dim1)
      and join-predicate(facts, dim2)
      and selection-predicate(dim1)
      and selection-predicate(dim2)
group by c1, c2
order by c1, c2
```

Result

Month	product	Sum of proceeds	Sum of quantity
February	pasta	110.000	45.000
March	pasta	95.000	50.000
April	pasta	105.000	51.000

Operations over multi-dimensional data

- ***Roll up*** — aggregates data
 - Sums up the sale quantity over last year per each region and product category
- ***Drill down*** — disaggregates data
 - For one particular product category in a region, “unrolls” and shows in detail the sale quantities of each day in each shop
- ***Slice & dice*** — selection and projection
- ***Pivot*** — change the orientation of the data cube

Drill Down: adding one dimension (**Zone**)

Month	Product	Zone	Sum of quantity
February	pasta	north	15.000
February	pasta	east	17.000
February	pasta	west	13.000
March	pasta	north	18.000
March	pasta	east	18.000
March	pasta	west	14.000
April	pasta	north	18.000
April	pasta	east	17.000
April	pasta	west	16.000

Roll-up: removing one dimension (Month)

Product	Zone	Sum of quantity
pasta	north	51.000
pasta	east	52.000
pasta	west	43.000

Aggregate queries

- **Examples:**

- Total proceeds for each product category in each shop in each day
- Total monthly proceeds in each shop
- Total monthly proceeds for each product category in each shop
- Average monthly proceeds for each category (calculated over all shops)

Aggregates in SQL: **data cube**

- Expresses all possible aggregations of the tuples of a table
- Uses a new purpose-specific *polymorphic* value: **ALL**

Data cube in SQL

```
select Model, Year,  
       Color, sum( Quantity )  
from Sales  
where Model in ('Fiat', 'Ford')  
      and Color = 'Red'  
      and Year between 1994 and 1995  
group by Model, Year, Color  
with cube
```

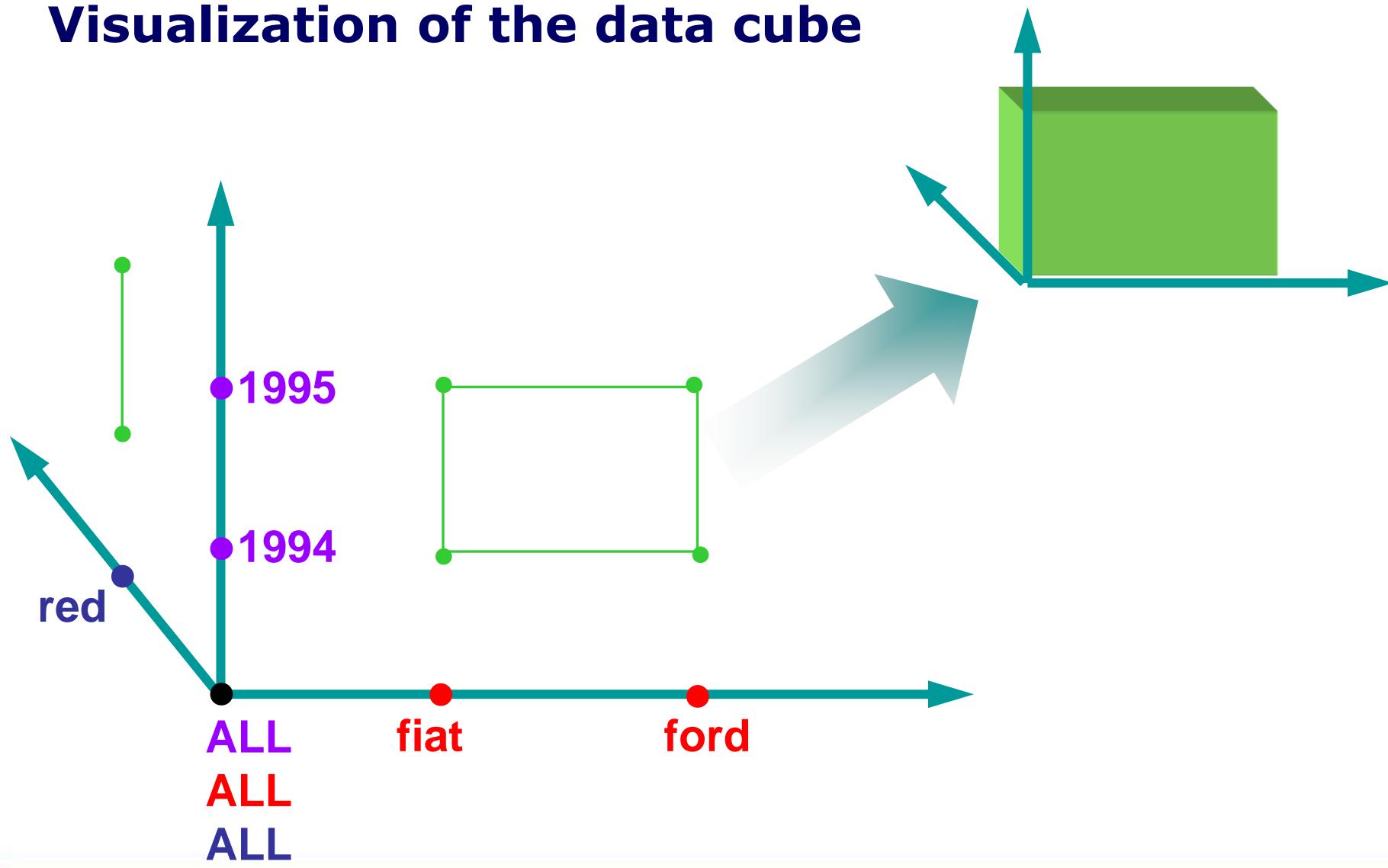
Relevant facts

Model	Year	Color	Quantity
fiat	1994	red	50
fiat	1995	red	85
ford	1994	red	80

Data cube results:

model	year	color	sum (quantity)
fiat	1994	red	50
fiat	1995	red	85
fiat	1994	ALL	50
fiat	1995	ALL	85
fiat	ALL	red	135
fiat	ALL	ALL	135
ford	1994	red	80
ford	1994	ALL	80
ford	ALL	red	80
ford	ALL	ALL	80
ALL	1994	red	130
ALL	1995	red	85
ALL	ALL	red	215
ALL	1994	ALL	130
ALL	1995	ALL	85
ALL	ALL	ALL	215

Visualization of the data cube



Roll up in SQL

```
select Model, Year,  
       Color, sum( Quantity )  
from Sales  
where Model in ('Fiat', 'Ford')  
    and Color = 'Red'  
    and Year between 1994 and 1995  
group by Model, Year, Color  
with roll up
```

Roll up results:

Model	Year	Color	sum(Quantity)
fiat	1994	red	50
fiat	1995	red	85
ford	1994	red	80
fiat	1994	ALL	50
fiat	1995	ALL	85
ford	1994	ALL	80
fiat	ALL	ALL	135
ford	ALL	ALL	80
ALL	ALL	ALL	215

Typical Size of a Data Warehouse

time: 730 days

shops: 300

products: 30.000

daily sales (avg number of sales of a product in a daily time slot for each shop): 3.000

offers: at most one per product on sale

sales: $730 * 300 * 3000 * 1 = 657 \text{ millions}$

Size fact table: $657 \text{ millions} * 8 \text{ attributes} * 4 \text{ Byte} = 21 \text{ GB}$

Data warehouse design

Data Warehouse design

- The design of a data warehouse differs from the design of an operational database
 - Data have different features
 - Constrained by existing databases
 - Driven by different design criteria
- Emphasis on abstraction and conceptual clarity
 - Few entities
 - Wide coverage
- Main activities
 - analysis — of existing data sources
 - integration
 - design — conceptual, logical and physical

Integration of data sources

- Data source integration is the merging of data represented in multiple sources into a unique global data base that represents the global information assets of the organization
- Main goal of integration is the identification of all the portions of the distinct data sources that refer to the same aspect of the described reality, to unify its representation
- The approach is directed toward the identification, analysis and resolution of conflicts — terminological, structural, and codification

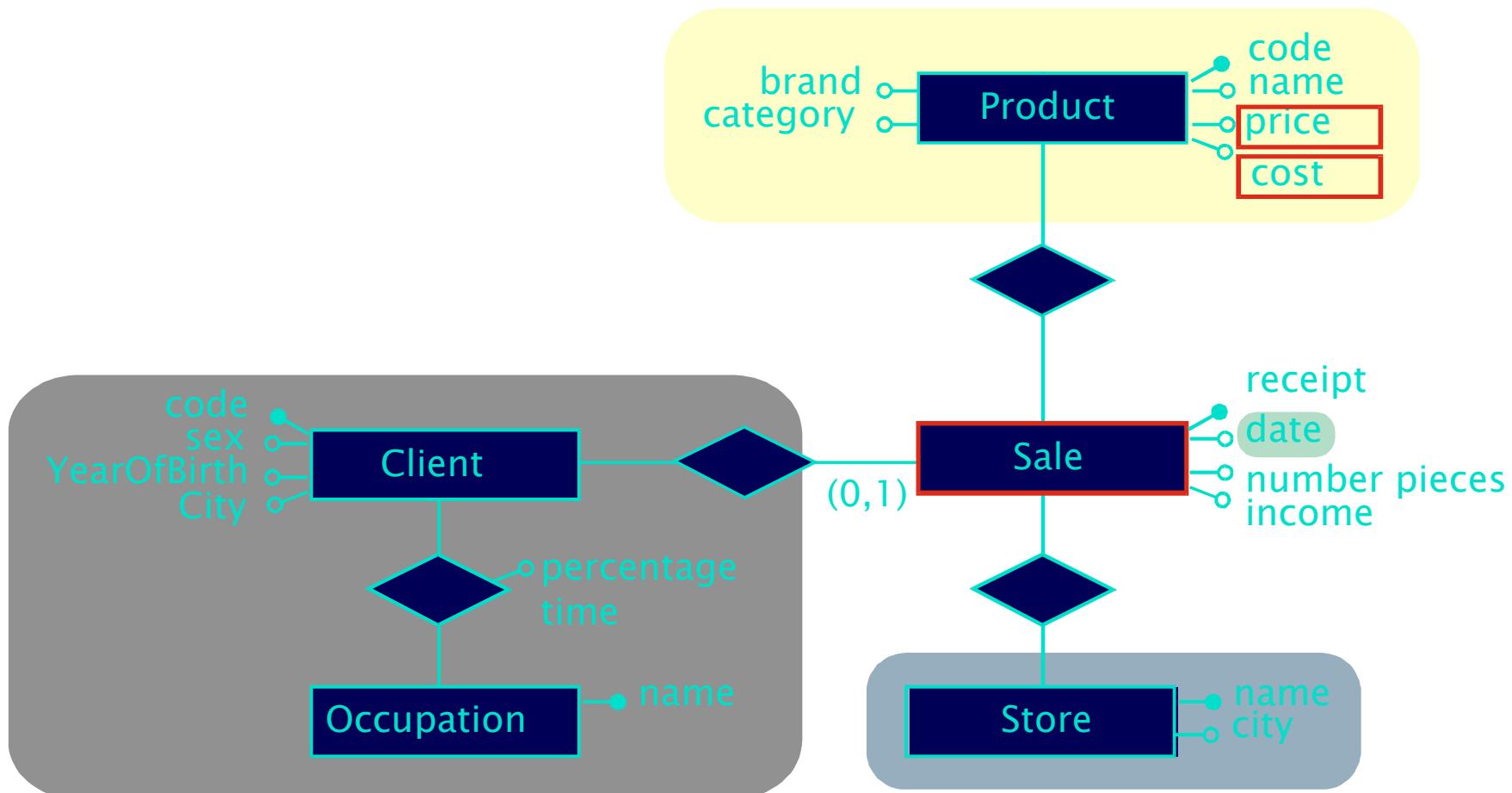
Examples of conflicts

- Conflicts associated with the codification of information
 - A «sex» attribute can be represented as:
 - With a single char — M/F
 - With a digit — 0/1
 - Implicit in the fiscal code
 - Not represented
 - First and last names of a person
 - “Mario”, “Rossi”
 - “Mario Rossi”
 - “Rossi, Mario”
 - “Rossi, M.”

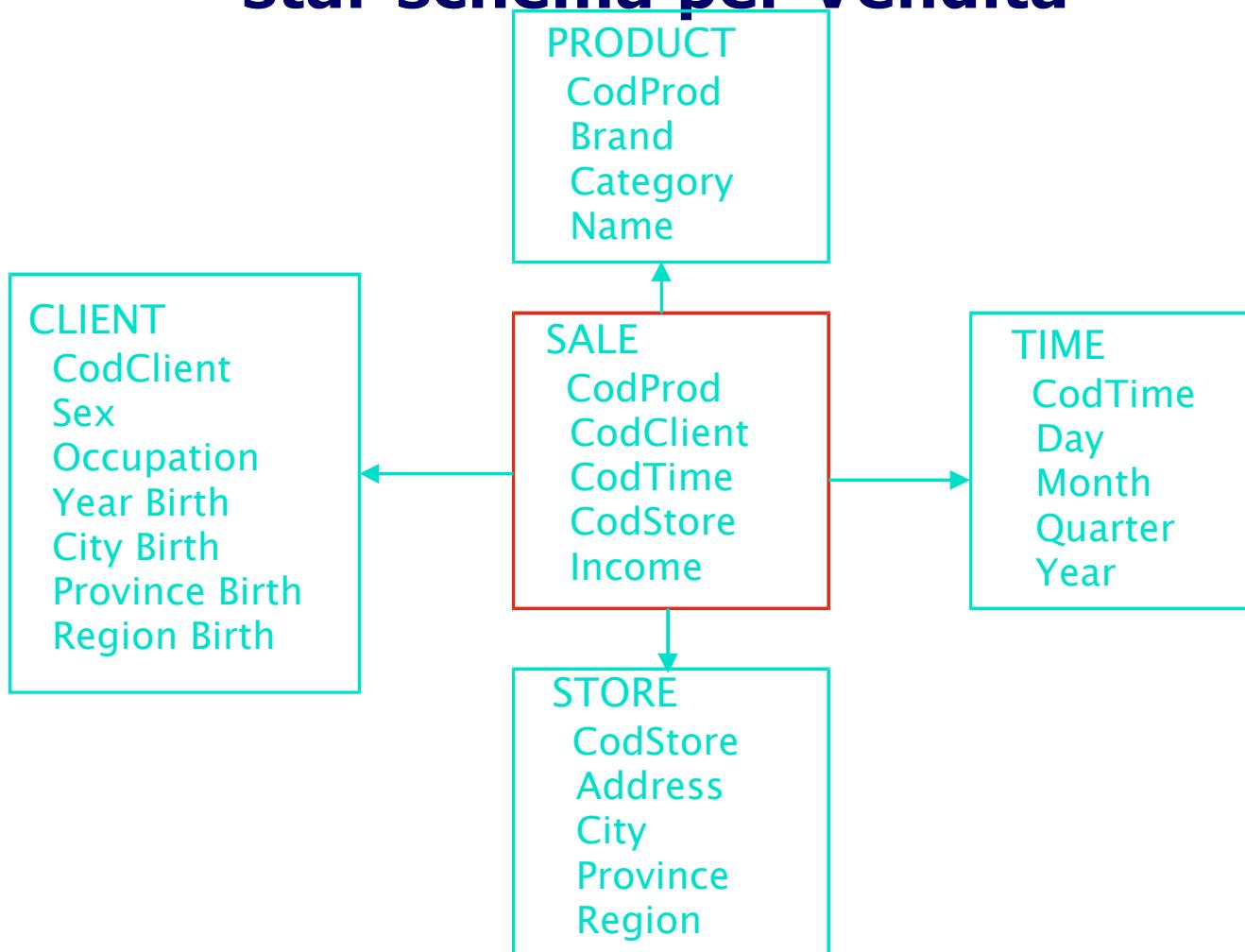
Identification of data marts

- Usually: several data marts present in the data warehouse
- Activities
 - Identification of facts, measures and dimensions
 - Restructuring of the conceptual schema
 - Representation of facts by way of entities
 - Identification of new dimensions
 - Refinement of the dimension levels
 - Derivation of a dimensional graph (star or snow-flake, depending on the scenario)
 - Logical and physical design of data marts, and of the population mechanism

Identification of facts and dimensions



Progettazione logica: star schema per Vendita



Advanced Databases

7

Data Warehouses

Technical aspects

Classification of OLAP System

- MOLAP (Multi-dimensional OLAP)
 - as alternative to
- ROLAP (Relational OLAP)
 - MOLAP: the internal data storage is not relational, so as to guarantee better performance
 - ROLAP: the relational storage guarantees the capability of managing large volumes of data

Specific OLAP Technologies

- **Bitmap Indexes**

- Allow for efficient evaluation of OR and AND combinations of simple comparison predicates

- **Join Indexes**

- Pre-computed joins between the table of facts and the tables representing the dimensions

- **Materialized views**

- Those views are pre-computed, which can be used to answer most frequently asked queries

Advanced Databases

7

Data Warehouses

Data Mining

Data mining

- Objective
 - Extract information *hidden* into data so as to support strategic decisions
- An inter-disciplinary task (and subject)
 - Statistics
 - Algorithmics
 - Neural networks
 - Fractals
 - ...

Applications of Data Mining

- Market analysis
 - Which products are purchased together or one before another? (basket analysis)
- Analysis of behaviours
 - Identify fraudulent credit card usage
- Forecasts
 - Foreseeing the cost of medical treatments
- Control
 - Industrial production errors

An example: sales analysis

Transaction	Date	Item	Qty	Price
1	12/17/95	ski-pants	1	140 €
1	12/17/95	ski-boots	1	180 €
2	12/18/95	T-shirt	1	25 €
2	12/18/95	jacket	1	300 €
2	12/18/95	ski-boots	1	70 €
3	12/18/95	jacket	1	300 €
4	12/19/95	T-shirt	3	25 €
4	12/19/95	jacket	1	300 €

Association Rules

- Association rules look for *regularity* within data
 - When a customer buys ski-boots, she also buys skis
- Structure of association rules:

Body* \Rightarrow *Head

- *Body*: premise of the rule
- *Head*: consequence of the rule

An example of Association Rule

Diaper \Rightarrow Beer

- 2% of all transactions contain both items
- 30% of transactions containing Diaper also contain Beer

Characteristics of Association Rules

- **Support**

- Probability that both Head and Body are in the same transaction [$P(H,B)$]

- **Confidence**

- Probability that the Head is in a transaction t, given that the Body **is** in t [$P(H|B)$, *conditional probability*]

- **Problem statement**

- Extract from a dataset all association rules with support and confidence over given thresholds

Examples of association rules

Body	Head	Support	Confidence
ski-pants	ski-boots	0.25	1
ski-boots	ski-pants	0.25	1
T-shirt	ski-boots	0.25	0.5
T-shirt	jacket	0.25	1
ski-boots	T-shirt	0.25	0.5
ski-boots	jacket	0.25	1
jacket	T-shirt	0.5	0.66
jacket	ski-boots	0.25	0.33
{ T-shirt, ski-boots }	jacket	0.25	1
{ T-shirt, jacket }	ski-boots	0.25	0.5
{ ski-boots, jacket }	T-shirt	0.25	1

Other Examples

- Items sold in the same special offer
- Items frequently purchased together in summer but not in winter
- Items frequently purchased together as in the shop they are arranged in a particular layout (adjacent, near, ...)
- Items purchased in consecutive transactions by the same customer

Sequential Patterns

- Input dataset:
 - All the transactions of a given customer
- Objective:
 - Find those sequences of items which are frequently contained into corresponding sequences of transactions, such that the frequency is over a given threshold

Examples

- “5% of customers bought a CD player in a transaction and some CDs in the following two transactions”
- “10% of the purchases of a television set is followed by the purchase of a video-recorder”
- Applications
 - Measure of the customer satisfaction
 - Special offers tailored for specific customer classes
 - Medicine (sequences of symptoms \Rightarrow disease)

Discretization

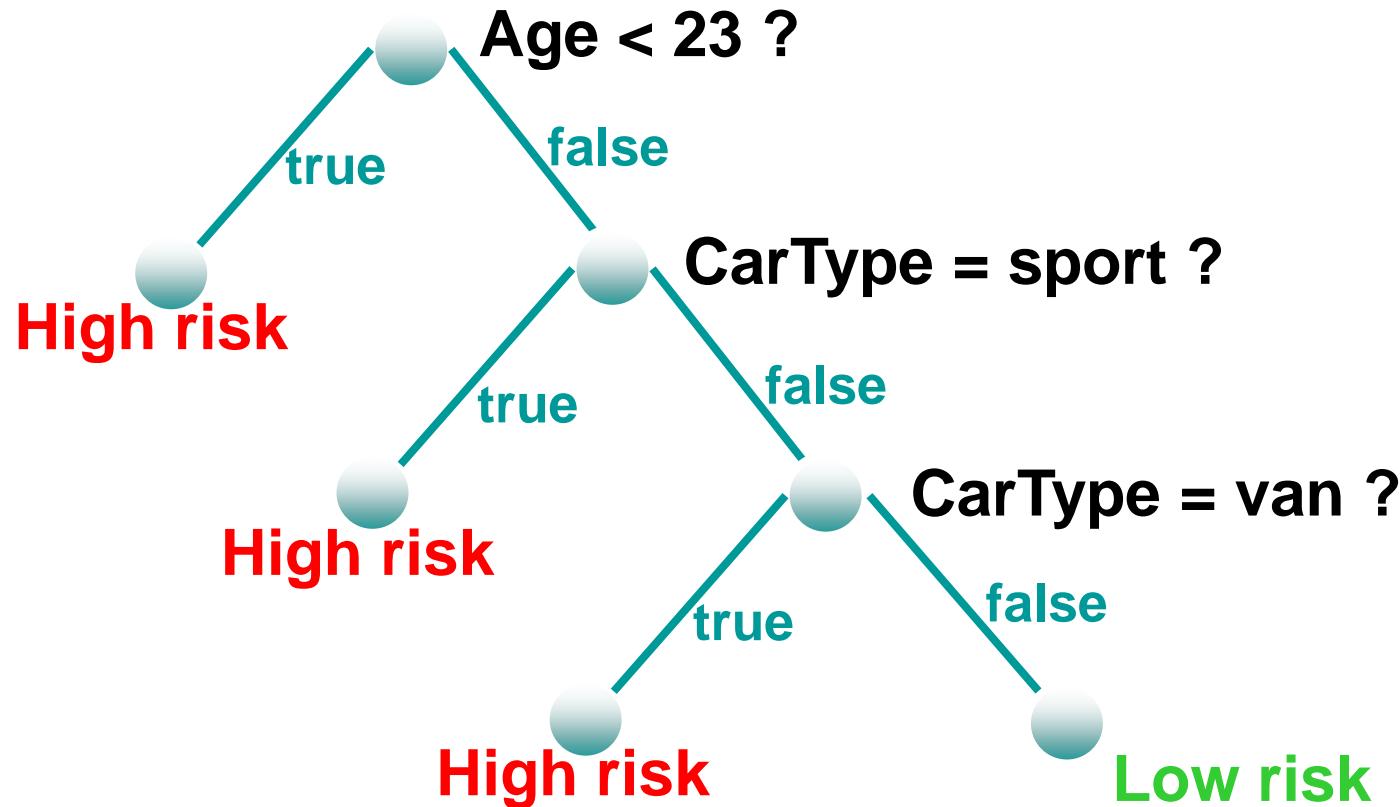
- A continuous domain can be represented by means of a sequence of suitable intervals
 - Example: blood-pressure
 - High: >250
 - Medium: $>130, <250$
 - Low: <130
- *Objective*: find the correlation between the risk of infarction and blood-pressure with a given statistical significance
- *Advantages*:
 - Compact value representation
 - Determination of critical values
 - Facilitation of future data analysis

Classification

- Cataloguing a fact, concept, or phenomenon into a pre-defined class
- The phenomenon is described by elementary facts (atomic data, within **tuples**)
- The classifier is constructed and trained over a set of training data (**training set**)
- Classifiers are represented as **decision-trees**

Example: identify risky policies

POLICY (DrivingLicense, Age, CarType)



Privacy-Aware Data Management and Information Security

Pierangela Samarati

Dipartimento di Informatica

Università degli Studi di Milano

pierangela.samarati@unimi.it

Material used by Stefano Paraboschi for
“Advanced Database Management”

Motivation

Ubiquitous, pervasive, open systems

- new ways to establish and manage identities
- new opportunities to enhance privacy
- new needs for privacy
- new risks for privacy

Scenario

Integration and sharing of different services of different parties in an open environment

- involving dynamic collaboration and process integration
- involving sharing of information from different authorities
- involving multiplicity and heterogeneity of entities and security specifications
- in a possible mobile/dynamic environment
- parties may be unknown
- lots of data communicated, collected, shared (information explosion!)

Outline

- Data Privacy: k -anonymity
- Privacy in Data Outsourcing
- Data Fragmentation and Encryption
- Authorization Enforcement in Collaborative Distributed Scenarios

Data Privacy: k -anonymity

Data collection and disclosure

- Internet provides unprecedented opportunities for the collection and sharing of privacy-sensitive information from and about users
- Information about users is collected every day
- Users have very strong concerns about the privacy of their personal information
- Protecting privacy requires the investigation of different issues, including the problem of protecting released information against inference and linking attacks which are becoming easier and easier because of the increased information availability and ease of access

Statistical DBMS vs statistical data

Often statistical data (or data for statistical purpose) are released

- **statistical DBMS [AW-89]**
 - the DBMS responds only to statistical queries
 - need run time checking to control information (indirectly) released
- **statistical data [CDFS-07b]**
 - publish statistics
 - control on indirect release performed before publication

Disclosure risk

Statistical data, even if ‘anonymized’, can be used to infer information that was not intended for disclosure

Disclosure can:

- occur based on the released data alone
- result from combination of the released data with publicly available information
- be possible only through combination of the released data with detailed external data sources that may or may not be available to the general public

When releasing data, the disclosure risk of sensitive information should be very low

Macrodata vs microdata

- In the past data were mainly released in tabular form (**macrodata**) and through statistical databases
- Today many situations require that the specific stored data themselves, called **microdata**, be released
 - increased flexibility and availability of information for the users
- Microdata are subject to a greater risk of privacy breaches
- The main requirements that must be taken into account are:
 - identity disclosure protection
 - attribute disclosure protection
 - inferential disclosure protection

Macrodata

Macrodata tables can be classified into the following two groups (types of tables)

- Count/Frequency. Each cell of the table contains the number of respondents (count) or the percentage of respondents (frequency) that have the same value over all attributes of analysis associated with the table
- Magnitude data. Each cell of the table contains an aggregate value of a *quantity of interest* over all attributes of analysis associated with the table

Count table – Example

Two-dimensional table showing the number of beneficiaries by county and size of benefit

County	Benefit						Total
	\$0-19	\$20-39	\$40-59	\$60-79	\$80-99	\$100+	
A	2	4	18	20	7	1	52
B	-	-	7	9	-	-	16
C	-	6	30	15	4	-	55
D	-	-	2	-	-	-	2

Magnitude table – Example

Average number of days spent in the hospital by respondents with a disease

	Hypertension	Obesity	Chest Pain	Short Breath	Tot
M	2	8.5	23.5	3	37
F	3	30.5	0	5	38.5
Tot	5	39	23.5	8	75.5

Microdata table – Example

Records about delinquent children in county Alfa

N	Child	County	Educ. HH	Salary HH	Race HH
1	John	Alfa	very high	201	black
2	Jim	Alfa	high	103	white
3	Sue	Alfa	high	77	black
4	Pete	Alfa	high	61	white
5	Ramesh	Alfa	medium	72	white
6	Dante	Alfa	low	103	white
7	Virgil	Alfa	low	91	black
8	Wanda	Alfa	low	84	white
9	Stan	Alfa	low	75	white
10	Irmi	Alfa	low	62	black
11	Renee	Alfa	low	58	white
12	Virginia	Alfa	low	56	black
13	Mary	Alfa	low	54	black
14	Kim	Alfa	low	52	white
15	Tom	Alfa	low	55	black
16	Ken	Alfa	low	48	white
17	Mike	Alfa	low	48	white
18	Joe	Alfa	low	41	black
19	Jeff	Alfa	low	44	black
20	Nancy	Alfa	low	37	white

Information disclosure

Disclosure relates to attribution of sensitive information to a respondent (an individual or organization)

There is disclosure when:

- a respondent is identified from released data (**identity disclosure**)
- sensitive information about a respondent is revealed through the released data (**attribute disclosure**)
- the released data make it possible to determine the value of some characteristic of a respondent even if no released record refers to the respondent (**inferential disclosure**)

Identity disclosure

It occurs if a third party can identify a respondent from the released data

Revealing that an individual is a respondent in a data collection may or may not violate confidentiality requirements

- Macrodata: revealing identity is generally not a problem, unless the identification leads to divulging confidential information (attribute disclosure)
- Microdata: identification is generally regarded as a problem, since microdata records are detailed; identity disclosure usually implies in this case also attribute disclosure

Attribute disclosure

It occurs when confidential information about a respondent is revealed and can be attributed to it; confidential information may be:

- revealed exactly
- closely estimated

Inferential disclosure

It occurs when information can be inferred with high confidence from statistical properties of the released data

E.g., the data may show a high correlation between income and purchase price of home. As purchase price of home is typically public information, a third party might use this information to infer the income of a respondent

Inference disclosure not always represents a risk:

- statistical data are released for enabling users to infer and understand relationships between variables
- inferences are designed to predict aggregate behavior, not individual attributes, and are then often poor predictors of individual data values

Restricted data and restricted access (1)

The choice of statistical disclosure limitation methods depends on the nature of the data products whose confidentiality must be protected

Some microdata include explicit identifiers (e.g., name, address, or Social Security number)

Removing such identifiers is a first step in preparing for the release of microdata for which the confidentiality of individual information must be protected

Restricted data and restricted access (2)

Confidentiality can be protected by:

- restricting the amount of information in the released tables (restricted data)
- imposing conditions on access to the data products (restricted access)
- some combination of these two strategies

Disclosure protection techniques

The protection techniques include:

- **sampling**: data confidentiality is protected by conducting a sample survey rather than a census
- **special rules**: designed for specific tables, they impose restrictions on the level of detail that can be provided in a table
- **threshold rule**: rules that protect sensitive cells, e.g.,
 - cell suppression
 - random rounding
 - controlled rounding
 - confidentiality edit

The anonymity problem

- The amount of privately owned records that describe each citizen's finances, interests, and demographics is increasing every day
- These data are de-identified before release, that is, any explicit identifier (e.g., SSN) is removed
- De-identification is not sufficient
- Most municipalities sell population registers that include the identities of individuals along with basic demographics
- These data can then be used for linking identities with de-identified information \Longrightarrow **re-identification**

The anonymity problem – Example

SSN	Name	Race	Date of birth	Sex	ZIP	Marital status	Disease
		asian	64/04/12	F	94142	divorced	hypertension
		asian	64/09/13	F	94141	divorced	obesity
		asian	64/04/15	F	94139	married	chest pain
		asian	63/03/13	M	94139	married	obesity
		asian	63/03/18	M	94139	married	short breath
		black	64/09/27	F	94138	single	short breath
		black	64/09/27	F	94139	single	obesity
		white	64/09/27	F	94139	single	chest pain
		white	64/09/27	F	94141	widow	short breath

Name	Address	City	ZIP	DOB	Sex	Status
.....
.....
Sue J. Doe	900 Market St.	San Francisco	94142	64/04/12	F	divorced
.....

Classification of attributes in a microdata table

The attributes in the original microdata table can be classified as:

- **identifiers.** Attributes that uniquely identify a microdata respondent (e.g., SSN uniquely identifies the person with which is associated)
- **quasi-identifiers.** Attributes that, in combination, can be linked with external information to reidentify all or some of the respondents to whom information refers or reduce the uncertainty over their identities (e.g., DoB, ZIP, and Sex)
- **confidential.** Attributes of the microdata table that contain sensitive information (e.g., Disease)
- **non confidential.** Attributes that the respondents do not consider sensitive and whose release do not cause disclosure

Re-identification

A study performed in 2000 reported that the US population was uniquely identifiable by:

- year of birth, 5-digit ZIP code: 0,2%
- year of birth, county: 0,0%
- year and month of birth, 5-digit ZIP code: 4,2%
- year and month of birth, county: 0,2%
- year, month, and day of birth, 5-digit ZIP code: 63,3%
- year, month, and day of birth, county: 14,8%

Factors contributing to disclosure risk (1)

Possible sources of the disclosure risk of microdata

- Existence of high visibility records. Some records on the file may represent respondents with unique characteristics such as very unusual jobs (e.g., movie star) or very large incomes
- Possibility of matching the microdata with external information. There may be individuals in the population who possess a unique or peculiar combination of the characteristic variables on the microdata
 - if some of those individuals happen to be chosen in the sample of the population, there is a disclosure risk
 - note that the identity of the individuals that have been chosen should be kept secret

Factors contributing to disclosure risk (2)

The possibility of linking or its precision increases with:

- the existence of a high number of common attributes between the microdata table and the external sources
- the accuracy or resolution of the data
- the number of outside sources, not all of which may be known to the agency releasing the microdata

Factors contributing to decrease the disclosure risk (1)

- A microdata table often contains a subset of the whole population
 - this implies that the information of a specific respondent, which a malicious user may want to know, may not be included in the microdata table
- The information specified in microdata tables released to the public is not always up-to-date (often at least one or two-year old)
 - the values of the attributes of the corresponding respondents may have been changed in the meanwhile
 - the age of the external sources of information used for linking may be different from the age of the information contained in the microdata table

Factors contributing to decrease the disclosure risk (2)

- A microdata table and the external sources of information naturally contain noise that decreases the ability to link the information
- A microdata table and the external sources of information can contain data expressed in different forms thus decreasing the ability to link information

Measures of risk

Measuring the disclosure risk requires considering

- the probability that the respondent for whom an intruder is looking for is represented on both the microdata and some external file
- the probability that the matching variables are recorded in a linkable way on the microdata and on the external file
- the probability that the respondent for whom the intruder is looking for is unique (or peculiar) in the population of the external file

The percentage of records representing respondents who are unique in the population (**population unique**) plays a major role in the disclosure risk of microdata (with respect to the specific respondent)

Note that each population unique is a sample unique; the vice-versa is not true

k -anonymity [S-01] (1)

- k -anonymity, together with its enforcement via generalization and suppression, has been proposed as an approach to protect respondents' identities while releasing truthful information
- k -anonymity tries to capture the following requirement:
 - the released data should be indistinguishably related to no less than a certain number of respondents
- Quasi-identifier: Set of attributes that can be exploited for linking (whose release must be controlled)

k -anonymity (2)

- Basic idea: translate the k -anonymity requirement on the released data
 - each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k respondents
- In the released table the respondents must be indistinguishable (within a given set) with respect to a set of attributes
- k -anonymity requires that each quasi-identifier value appearing in the released table must have at least k occurrences
 - sufficient condition for the satisfaction of k -anonymity requirement

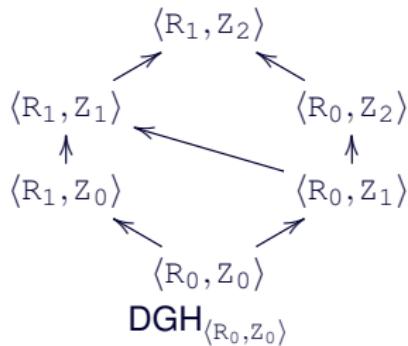
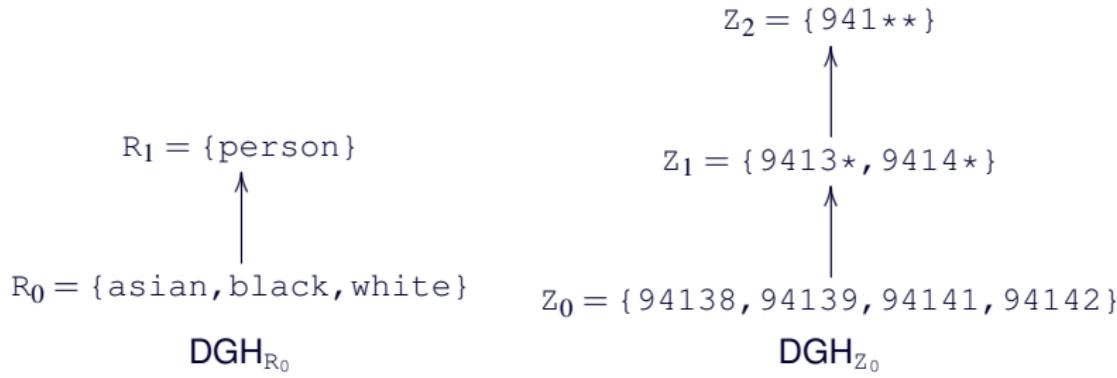
Generalization and suppression

- **Generalization.** The values of a given attribute are substituted by using more general values. Based on the definition of a generalization hierarchy
 - Example: consider attribute ZIP code and suppose that a step in the corresponding generalization hierarchy consists in suppressing the least significant digit in the ZIP code
With one generalization step: 20222 and 20223 become 2022*; 20238 and 20239 become 2023*
- **Suppression.** It is a well-known technique that consists in protecting sensitive information by removing it
 - the introduction of suppression can reduce the amount of generalization necessary to satisfy the k -anonymity constraint

Domain generalization hierarchy

- A generalization relationship \leq_D defines a mapping between domain D and its generalizations
- Given two domains $D_i, D_j \in \text{Dom}$, $D_i \leq_D D_j$ states that the values in domain D_j are generalizations of values in D_i
- \leq_D implies the existence, for each domain D , of a domain generalization hierarchy $\text{DGH}_D = (\text{Dom}, \leq_D)$:
 - $\forall D_i, D_j, D_z \in \text{Dom}$:
$$D_i \leq_D D_j, D_i \leq_D D_z \implies D_j \leq_D D_z \vee D_z \leq_D D_j$$
 - all maximal elements of Dom are singleton
- Given a domain tuple $DT = \langle D_1, \dots, D_n \rangle$ such that $D_i \in \text{Dom}$, $i = 1, \dots, n$, the domain generalization hierarchy of DT is $\text{DGH}_{DT} = \text{DGH}_{D_1} \times \dots \times \text{DGH}_{D_n}$

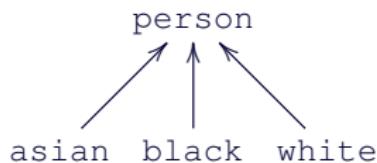
Examples of domain generalization hierarchies



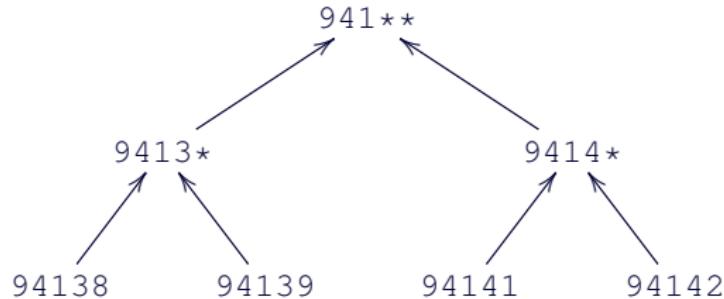
Value generalization hierarchy

- A value generalization relationship \leq_V associates with each value in domain D_i a unique value in domain D_j , direct generalization of D_i
- \leq_V implies the existence, for each domain D , of a value generalization hierarchy VGH_D
- VGH_D is a tree
 - the leaves are the values in D
 - the root (i.e., the most general value) is the value in the maximum element in DGH_D

Examples of value generalization hierarchies



VGH_{R_0}



VGH_{Z_0}

Generalized table with suppression

Let T_i and T_j be two tables defined on the same set of attributes. Table T_j is said to be a generalization (with tuple suppression) of table T_i , denoted $T_i \preceq T_j$, if:

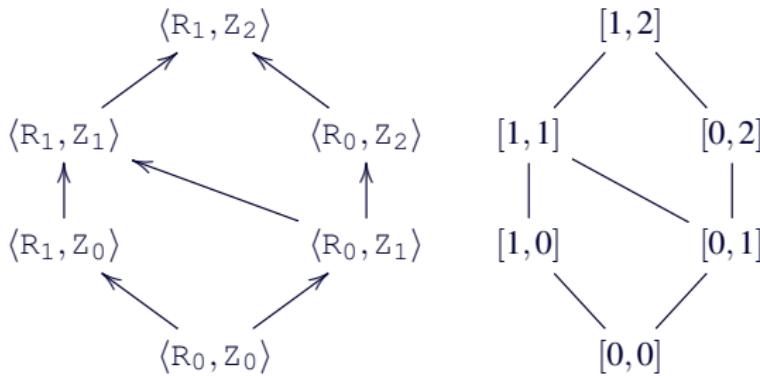
1. $|T_j| \leq |T_i|$;
2. the domain $\text{dom}(A, T_j)$ of each attribute A in T_j is equal to, or a generalization of, the domain $\text{dom}(A, T_i)$ of attribute A in T_i ;
3. it is possible to define an injective function associating each tuple t_j in T_j with a tuple t_i in T_i , such that the value of each attribute in t_j is equal to, or a generalization of, the value of the corresponding attribute in t_i .

Generalized table with suppression – Example

Race	ZIP
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

k -minimal generalization with suppression (1)

- Distance vector. Let $T_i(A_1, \dots, A_n)$ and $T_j(A_1, \dots, A_n)$ be two tables such that $T_i \preceq T_j$. The distance vector of T_j from T_i is the vector $DV_{i,j} = [d_1, \dots, d_n]$, where each d_z , $z = 1, \dots, n$, is the length of the unique path between $\text{dom}(A_z, T_i)$ and $\text{dom}(A_z, T_j)$ in the domain generalization hierarchy DGH_{D_z} .



k -minimal generalization with suppression (2)

Let T_i and T_j be two tables such that $T_i \preceq T_j$, and let MaxSup be the specified threshold of acceptable suppression. T_j is said to be a k -minimal generalization of table T_i iff:

1. T_j satisfies k -anonymity enforcing minimal required suppression, that is, T_j satisfies k -anonymity and $\forall T_z : T_i \preceq T_z, DV_{i,z} = DV_{i,j}$, T_z satisfies k -anonymity $\implies |T_j| \geq |T_z|$
2. $|T_i| - |T_j| \leq \text{MaxSup}$
3. $\forall T_z : T_i \preceq T_z$ and T_z satisfies conditions 1 and 2 $\implies \neg(DV_{i,z} < DV_{i,j})$

Examples of 2-minimal generalizations

MaxSup=2

Race: R_0	ZIP: Z_0	Race: R_1	ZIP: Z_0	Race: R_0	ZIP: Z_1
asian	94142			asian	9414*
asian	94141	person	94141	asian	9414*
asian	94139	person	94139	asian	9413*
asian	94139	person	94139	asian	9413*
asian	94139	person	94139	asian	9413*
black	94138			black	9413*
black	94139	person	94139	black	9413*
white	94139	person	94139		
white	94141	person	94141		

PT

$\text{GT}_{[1,0]}$

$\text{GT}_{[0,1]}$

Computing a preferred generalization

Different preference criteria can be applied in choosing a preferred minimal generalization, among which:

- **minimum absolute distance** prefers the generalization(s) with the smallest absolute distance, that is, with the smallest total number of generalization steps (regardless of the hierarchies on which they have been taken)
- **minimum relative distance** prefers the generalization(s) with the smallest relative distance, that is, that minimizes the total number of relative steps (a step is made relative by dividing it over the height of the domain hierarchy to which it refers)
- **maximum distribution** prefers the generalization(s) with the greatest number of distinct tuples
- **minimum suppression** prefers the generalization(s) that suppresses less tuples, that is, the one with the greatest cardinality

Classification of k -anonymity techniques (1)

Generalization and suppression can be applied at different levels of granularity

- Generalization can be applied at the level of single column (i.e., a generalization step generalizes all the values in the column) or single cell (i.e., for a specific column, the table may contain values at different generalization levels)
- Suppression can be applied at the level of row (i.e., a suppression operation removes a whole tuple), column (i.e., a suppression operation obscures all the values of a column), or single cells (i.e., a k -anonymized table may wipe out only certain cells of a given tuple/attribute)

Classification of k -anonymity techniques (2)

Generalization	Suppression			
	<i>Tuple</i>	<i>Attribute</i>	<i>Cell</i>	<i>None</i>
<i>Attribute</i>	AG_TS	AG_AS $\equiv AG_{}$	AG_CS	AG_ $\equiv AG_{AS}$
<i>Cell</i>	CG_TS not applicable	CG_AS not applicable	CG_CS $\equiv CG_{}$	CG_ $\equiv CG_{CS}$
<i>None</i>	_TS	_AS	_CS	— not interesting

2-anonymized tables wrt different models (1)

Race	DOB	Sex	ZIP
asian	64/04/12	F	94142
asian	64/09/13	F	94141
asian	64/04/15	F	94139
asian	63/03/13	M	94139
asian	63/03/18	M	94139
black	64/09/27	F	94138
black	64/09/27	F	94139
white	64/09/27	F	94139
white	64/09/27	F	94141

PT

Race	DOB	Sex	ZIP
asian	64/04	F	941**
asian	64/04	F	941**
asian	63/03	M	941**
asian	63/03	M	941**
black	64/09	F	941**
black	64/09	F	941**
white	64/09	F	941**
white	64/09	F	941**

AG_TS

2-anonymized tables wrt different models (2)

Race	DOB	Sex	ZIP
asian	*	F	*
asian	*	F	*
asian	*	F	*
asian	63/03	M	9413*
asian	63/03	M	9413*
black	64/09	F	9413*
black	64/09	F	9413*
white	64/09	F	*
white	64/09	F	*

AG_CS

Race	DOB	Sex	ZIP
asian	64	F	941**
asian	64	F	941**
asian	64	F	941**
asian	63	M	941**
asian	63	M	941**
black	64	F	941**
black	64	F	941**
white	64	F	941**
white	64	F	941**

AG_ ≡ AG_AS

2-anonymized tables wrt different models (3)

Race	DOB	Sex	ZIP
asian	64	F	941**
asian	64	F	941**
asian	64	F	941**
asian	63/03	M	94139
asian	63/03	M	94139
black	64/09/27	F	9413*
black	64/09/27	F	9413*
white	64/09/27	F	941**
white	64/09/27	F	941**

CG \equiv CG_CS

Race	DOB	Sex	ZIP

_TS

2-anonymized tables wrt different models (4)

Race	DOB	Sex	ZIP
asian	*	F	*
asian	*	F	*
asian	*	F	*
asian	*	M	*
asian	*	M	*
black	*	F	*
black	*	F	*
white	*	F	*
white	*	F	*

_AS

Race	DOB	Sex	ZIP
asian	*	F	*
asian	*	F	*
asian	*	F	*
asian	*	M	94139
asian	*	M	94139
*	64/09/27	F	*
*	64/09/27	F	94139
*	64/09/27	F	94139
*	64/09/27	F	*

_CS

Algorithms for computing a k -anonymous table

- The problem of finding minimal k -anonymous tables, with attribute generalization and tuple suppression, is computationally hard
- The majority of the exact algorithms proposed in literature have computational time exponential in the number of the attributes composing the quasi-identifier
 - when the number $|QI|$ of attributes in the quasi-identifier is small compared with the number n of tuples in the private table PT, these exact algorithms with attribute generalization and tuple suppression are practical
- Recently many exact algorithms for producing k -anonymous tables through attribute generalization and tuple suppression have been proposed

Algorithms for **AG_TS** and **AG_**

Samarati's algorithm [S-01] (1)

- Each path in DGH_{DT} represents a generalization strategy for PT
- We call **locally minimal generalization** the lowest node of each path satisfying k -anonymity
- Properties exploited by the algorithm:
 1. each k -minimal generalization is locally minimal with respect to a path (but the converse is not true)
 2. going up in the hierarchy the number of tuples that must be removed to guarantee k -anonymity decreases
- If there is no solution that guarantees k -anonymity suppressing less than MaxSup tuples at height h , there cannot exist a solution, with height lower than h that guarantees it

Samarati's algorithm (2)

- The algorithm adopts a *binary search* on the lattice of distance vectors:
 1. evaluate solutions at height $\lfloor h/2 \rfloor$
 2. if there exists at least a solution satisfying k -anonymity
 - then evaluates solutions at height $\lfloor h/4 \rfloor$
 - otherwise evaluates solutions at height $\lfloor 3h/4 \rfloor$
 3. until the algorithm reaches the lowest height for which there is a distance vector that satisfies k -anonymity
- To reduce the computational cost, it adopts a **distance vector matrix** that avoids the explicit computation of each generalized table

Samarati's algorithm – Example (1)

Distance vector matrix

Race: R ₀	ZIP: Z ₀
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

	t ₁	t ₂	t ₃ /t ₄ /t ₅	t ₆	t ₇	t ₈	t ₉
t ₁	[0, 0]	[0, 1]	[0, 2]	[1, 2]	[1, 2]	[1, 2]	[1, 1]
t ₂	[0, 1]	[0, 0]	[0, 2]	[1, 2]	[1, 2]	[1, 2]	[1, 0]
t ₆	[1, 2]	[1, 2]	[1, 1]	[0, 0]	[0, 1]	[1, 1]	[1, 2]
t ₇	[1, 2]	[1, 2]	[1, 0]	[0, 1]	[0, 0]	[1, 0]	[1, 2]
t ₈	[1, 2]	[1, 2]	[1, 0]	[1, 1]	[1, 0]	[0, 0]	[0, 2]
t ₉	[1, 1]	[1, 0]	[1, 2]	[1, 2]	[1, 2]	[0, 2]	[0, 0]

Samarati's algorithm – Example (2)

Compute solutions at height 0: $\text{GT}_{[0,0]}$

Race: R_0	ZIP: Z_0
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

The generalized table does not satisfy 2-anonymity

Samarati's algorithm – Example (3)

Suppose $k = 2$ and MaxSup=2.

Compute first solutions at height 1: $\text{GT}_{[1,0]}$ and $\text{GT}_{[0,1]}$

Race: R_1	ZIP: Z_0	Race: R_0	ZIP: Z_1
person	94142	asian	9414*
person	94141	asian	9414*
person	94139	asian	9413*
person	94139	asian	9413*
person	94139	asian	9413*
person	94138	black	9413*
person	94139	black	9413*
person	94139	white	9413*
person	94141	white	9414*

Both the generalized tables satisfy 2-anonymity

k -Optimize algorithm (1) [BA-05]

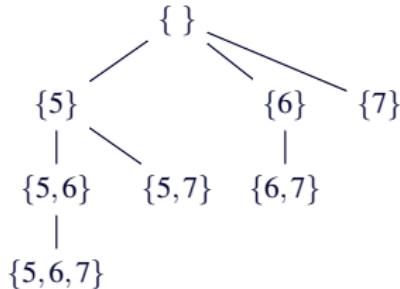
- Order attributes in QI and the values in their domains
- Associate an integer *index* value with each domain value, following the defined order

Race		ZIP	
$\langle [\text{asian}] \; [\text{black}] \; [\text{white}] \rangle$		$\langle [94138] \; [94139] \; [94141] \; [94142] \rangle$	
1	2	3	4
			5
			6
			7

- A generalization is the union of individual index values
- The least value in an attribute domain is omitted. E.g., {6} corresponds to:
 - Race: {1}, that is: $\langle [\text{asian or black or white}] \rangle$
 - ZIP: {4, 6}, that is: $\langle [94138 \text{ or } 94139], [94141 \text{ or } 94142] \rangle$
- Order of values within domains has impact on generalization

k -Optimize algorithm (2)

- k -Optimize builds a set enumeration tree over the set I of indexes



- The root node of the tree is the empty set
- The children of n are the sets obtained by appending a single element i of I to n , such that $\forall i' \in n, i > i'$
- Each node is associated with a cost that reflects both the amount of generalization and suppression of the anonymization represented by the node
 - ⇒ each tuple is associated with a cost that reflects the information loss associated with its generalization or suppression

k -Optimize algorithm (3)

- k -Optimize visits the tree (e.g., using a depth-first search) for searching the anonymization with lowest cost
- Since the number of nodes in the tree is $2^{|I|}$ the visit of the tree is not practical \Rightarrow pruning strategy to reduce computational cost
- Node n is pruned iff none of its descendants could be optimal
- This determination can be made by computing a lower bound on the cost of the nodes in the subtree rooted at n
 \Rightarrow if the lower bound is greater than the current best cost, node n is pruned

Incognito algorithm [LDR-05]

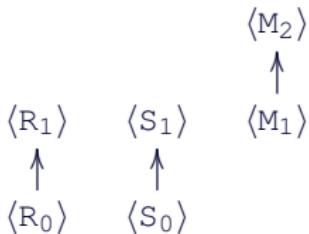
k -anonymity with respect to a proper subset of QI is a necessary (not sufficient) condition for k -anonymity with respect to QI

- Iteration 1: check k -anonymity for each attribute in QI , discarding generalizations that do not satisfy k -anonymity
- Iteration 2: combine the remaining generalizations in pairs and check k -anonymity for each couple obtained
...
- Iteration i : consider all the i -uples of attributes, obtained combining generalizations that satisfied k -anonymity at iteration $i - 1$. Discard non k -anonymous solutions
...
- Iteration $|QI|$ returns the final result

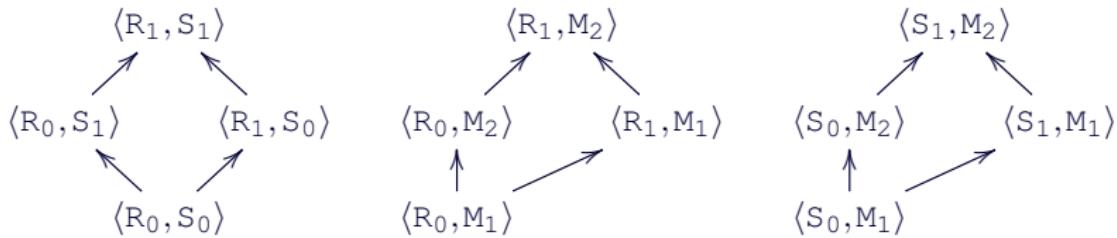
Incognito adopts a bottom-up approach for the visit of DGHs

Incognito – Example (1)

Iteration 1

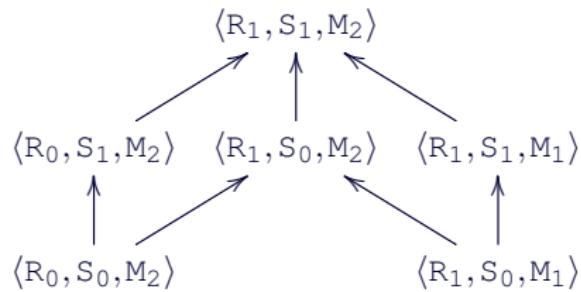


Iteration 2



Incognito – Example (2)

Iteration 3



Heuristic algorithms

- The exact algorithms have complexity exponential in the size of QI
- Heuristic algorithms have been proposed
 - [I-02]: based on genetic algorithms, it solves the k -anonymity problem using an incomplete stochastic search method
 - [W-04]: based on simulated annealing for finding locally minimal solutions, it requires high computational time and does not assure the quality of the solution
 - [FWY-05]: top-down heuristic to make a table to be released k -anonymous; it starts from the most general solution, and iteratively specializes some values of the current solution until the k -anonymity requirement is violated
- No bounds on efficiency and goodness of the solutions can be given
- Experimental results can be used to assess the quality of the solution retrieved

Algorithms for CS and CG

Mondrian multidimensional algorithm [LDR-06] (1)

- Each attribute in QI represents a dimension
- Each tuple in PT represents a point in the space defined by QI
- Tuples with the same QI value are represented by giving a multiplicity value to points
- The multi-dimensional space is partitioned by splitting dimensions such that each area contains at least k occurrences of point values
- All the points in a region are generalized to a unique value
- The corresponding tuples are substituted by the computed generalization

Mondrian multidimensional algorithm (2)

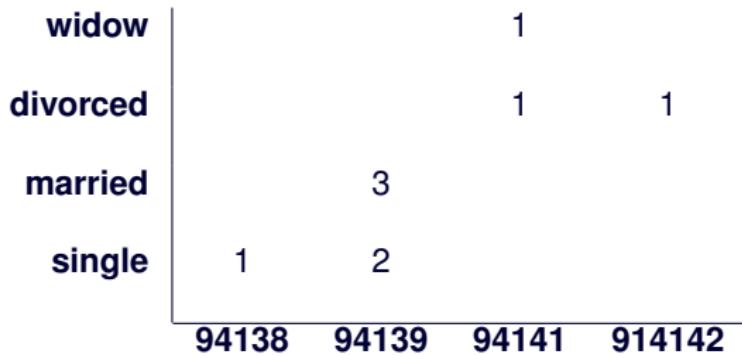
Mondrian algorithm is flexible and can operate

- on a different number of attributes
 - single-dimension
 - multi-dimension
- with different recoding (generalization) strategies
 - global recoding
 - local recoding
- with different partitioning strategies
 - strict (i.e., non-overlapping) partitioning
 - relaxed (i.e., potentially overlapping) partitioning
- using different metrics to determine how to split on each dimension

Mondrian multidimensional algorithm – Example (1)

Private table

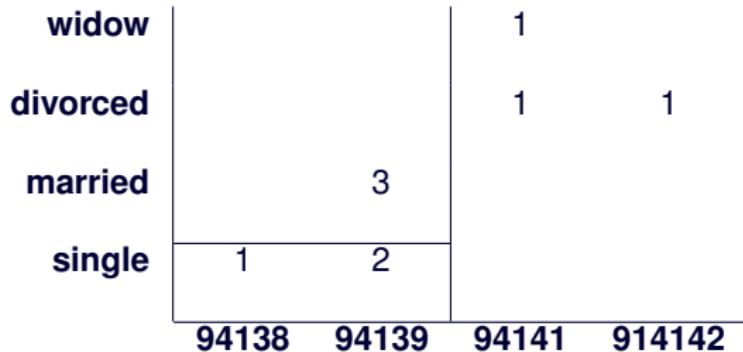
Marital status	ZIP
divorced	94142
divorced	94141
married	94139
married	94139
married	94139
single	94138
single	94139
single	94139
widow	94141



Mondrian multidimensional algorithm – Example (2)

3-anonymous table

Marital status	ZIP
divorced or widow	9414*
divorced or widow	9414*
married	94139
married	94139
married	94139
single	9413*
single	9413*
single	9413*
divorced or widow	9414*



Approximation algorithms

- Approximation algorithms for general and specific values of k (e.g., 1.5-approximation for 2-anonymity, and 2-approximation for 3-anonymity [AFKMPTZ-05b])
- Approximation algorithm for **_CS**
 - [MW-04]: $O(k \log(k))$ -approximation
 - [AFKMPTZ-05a]: with unbounded value of k , $O(k)$ -approximation solution
- Approximation algorithm for **CG_**
 - [AFKMPTZ-05b]: with unbounded value of k , $O(k)$ -approximation solution

k -anonymity revisited [GMT-08]

- *k -anonymity requirement:* Each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k respondents
- When generalization is performed at attribute level (**AG**) this is equivalent to require each quasi-identifier n-uple to have at least k occurrences
- When generalization is performed at cell level (**CG**) the existence of at least k occurrences is a sufficient but not necessary condition; a less stricter requirement would suffice
 1. For each sequence of values pt in $PT[q]$ there are at least k tuples in $T[q]$ that contain a sequence of values generalizing pt
 2. For each sequence of values t in $T[q]$ there are at least k tuples in $PT[q]$ that contain a sequence of values for which t is a generalization

k -anonymity revisited – Example

Race	ZIP
white	94138
black	94139
asian	94141
asian	94141
asian	94142

PT

Race	ZIP
person	9413*
person	9413*
asian	9414*
asian	9414*
asian	9414*

2-anonymity

Race	ZIP
person	9413*
person	9413*
asian	94141
asian	9414*
asian	9414*

2-anonymity (revisited)

Race	ZIP
person	9413*
person	9413*
asian	9414*
asian	9414*
asian	94142

no 2-anonymity

Race	ZIP
person	9413*
person	9413*
asian	94141
asian	94141
asian	9414*

Attribute Disclosure

2-anonymous table according to the AG_ model

k -anonymity is vulnerable to some attacks [MGK-06,S-01]

Race	DOB	Sex	ZIP	Disease
asian	64	F	941**	hypertension
asian	64	F	941**	obesity
asian	64	F	941**	chest pain
asian	63	M	941**	obesity
asian	63	M	941**	obesity
black	64	F	941**	short breath
black	64	F	941**	short breath
white	64	F	941**	chest pain
white	64	F	941**	short breath

Homogeneity of the sensitive attribute values

- All tuples with a quasi-identifier value in a k -anonymous table may have the same sensitive attribute value
 - an adversary knows that Carol is a black female and that her data are in the microdata table
 - the adversary can infer that Carol suffers from short breath

Race	DOB	Sex	ZIP	Disease
...
black	64	F	941**	short breath
black	64	F	941**	short breath
...

Background knowledge

- Based on prior knowledge of some additional external information
 - an adversary knows that Hellen is a white female and she is in the microdata table
 - the adversary can infer that the disease of Hellen is either chest pain or short breath
 - the adversary knows that the Hellen runs 2 hours a day and therefore that Hellen cannot suffer from short breath
⇒ the adversary infers that Hellen's disease is chest pain

Race	DOB	Sex	ZIP	Disease
...
white	64	F	941**	chest pain
white	64	F	941**	short breath

ℓ -diversity (1)

- A q -block (i.e., set of tuples with the same value for QI) in T is ℓ -diverse if it contains at least ℓ different “well-represented” values for the sensitive attribute in T
 - “well-represented” different definitions based on entropy or recursion (e.g., a q -block is ℓ -diverse if removing a sensitive value it remains $(\ell-1)$ -diverse)
- ℓ -diversity: an adversary needs to eliminate at least $\ell-1$ possible values to infer that a respondent has a given value

ℓ -diversity (2)

- T is ℓ -diverse if all its q -blocks are ℓ -diverse
 - ⇒ the homogeneity attack is not possible anymore
 - ⇒ the background knowledge attack becomes more difficult
- ℓ -diversity is monotonic with respect to the generalization hierarchies considered for k -anonymity purposes
- Any algorithm for k -anonymity can be extended to enforce the ℓ -diverse property

Skewness attack

ℓ -diversity leaves space to attacks based on the distribution of values inside q -blocks

- Skewness attack occurs when the distribution in a q -block is different than in the original population
- 20% populations suffers from diabetes; 75% of tuples in a q -block have diabetes
 \Rightarrow people in the q -block have higher probability of suffering from diabetes

Race	DOB	Sex	ZIP	Disease
black	64	F	941**	diabetes
black	64	F	941**	short breath
black	64	F	941**	diabetes
black	64	F	941**	diabetes

Similarity attack

- Similarity attack happens when a q -block has different but semantically similar values for the sensitive attribute

Race	DOB	Sex	ZIP	Disease
black	64	F	941**	stomach ulcer
black	64	F	941**	stomach ulcer
black	64	F	941**	gastritis

Group closeness [LLV-07]

- A q -block respects t -closeness if the distance between the distribution of the values of the sensitive attribute in the q -block and in the considered population is lower than t
- T respects t -closeness if all its q -blocks respect t -closeness
- t -closeness is monotonic with respect to the generalization hierarchies considered for k -anonymity purposes
- Any algorithm for k -anonymity can be extended to enforce the t -closeness property, which however might be difficult to achieve

- The consideration of the adversary's background knowledge (or external knowledge) is necessary when reasoning about privacy in data publishing
- External knowledge can be exploited for inferring sensitive information about individuals with high confidence
- Positive inference
 - a respondent has a given value (or a value within a restricted set)
- Negative inference
 - a respondent does not have a given value
- Existing approaches have mostly focused on positive inference

External knowledge (2)

- External knowledge may include:
 - similar datasets released by different organizations
 - instance-level information
 - ...
- Not possible to know a-priori what external knowledge the adversary possesses
- It is necessary to provide the data owner with a means to specify adversarial knowledge

External knowledge modeling [CLR-07]

- An adversary has knowledge about an individual (target) represented in a released table and knows the individual's QI values
 ⇒ goal: predict whether the target has a target sensitive value
- External knowledge modeled through a logical expression
- Three basic classes of expressions, representing knowledge about:
 - the target individual. Information that the adversary may know about the target individual
 - others. Information about individuals other than the target
 - same-value families. The knowledge that a group (or family) of individuals have the same sensitive value
- Other types of external knowledge may be identified.....

External knowledge – Example (1)

Name	DOB	Sex	ZIP	Disease
Alice	74/04/12	F	94142	aids
Bob	74/04/13	F	94141	flu
Carol	74/09/15	F	94139	flu
David	74/03/13	M	94139	aids
Elen	64/03/18	M	94139	flu
Frank	64/09/27	F	94138	short breath
George	64/09/27	F	94139	flu
Harry	64/09/27	M	94139	aids

Original table

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

Released table is 4-anonymized but

External knowledge – Example (2)

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

An adversary knows that Harry, born in 64 and living in area 94139, is in the table

External knowledge – Example (2)

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table



DOB	Sex	ZIP	Disease
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

An adversary knows that Harry, born in 64 and living in area 94139, is in the table

- ⇒ Harry belongs to the second group
- ⇒ Harry has aids with confidence 1/4

External knowledge – Example (3)

DOB	Sex	ZIP	Disease
-----	-----	-----	---------

64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

From another dataset, the adversary knows that George (who is in the table, is born in 64, and leaves in area 941**) has flu

External knowledge – Example (3)

DOB	Sex	ZIP	Disease
-----	-----	-----	---------

64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table



DOB	Sex	ZIP	Disease
-----	-----	-----	---------

64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

From another dataset, the adversary knows that George (who is in the table, is born in 64, and leaves in area 941**) has flu

⇒ Harry has aids with confidence 1/3

External knowledge – Example (4)

DOB	Sex	ZIP	Disease
-----	-----	-----	---------

64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

From personal knowledge, the adversary knows that Harry does not have short breath

External knowledge – Example (4)

DOB	Sex	ZIP	Disease
-----	-----	-----	---------

DOB	Sex	ZIP	Disease
-----	-----	-----	---------



64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

64	*	941**	flu
64	*	941**	aids

4-anonymized table

From personal knowledge, the adversary knows that Harry does not have short breath

⇒ Harry has aids with confidence 1/2

External knowledge – Example (5)

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

An adversary knows that Bob, born in 74 and living in area 94141, is in the table

External knowledge – Example (5)

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table



DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids

4-anonymized table

An adversary knows that Bob, born in 74 and living in area 94141, is in the table

⇒ Bob belongs to the first group

External knowledge – Example (5)

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table



DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids

4-anonymized table

An adversary knows that Bob, born in 74 and living in area 94141, is in the table

⇒ Bob belongs to the first group

Alice is Bob's wife and she is in the table

External knowledge – Example (5)

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table



DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids

4-anonymized table

An adversary knows that Bob, born in 74 and living in area 94141, is in the table

⇒ Bob belongs to the first group

Alice is Bob's wife and she is in the table

⇒ If Bob has aids, it is highly probable that Alice has aids

Some open issues

- Privacy metrics to model the privacy enjoyed by a data release
- New techniques to protect privacy (in addition to generalization and suppression)
- External knowledge and adversarial attacks
- Novel scenarios of data release, e.g.,
 - release of marginal tables [KG-06]
 - fragmented tables
- Evaluation of privacy vs utility
- Application of k -anonymity-based principles to other scenarios (e.g., social networks, location-based services)

Multi-dimensional Indexing

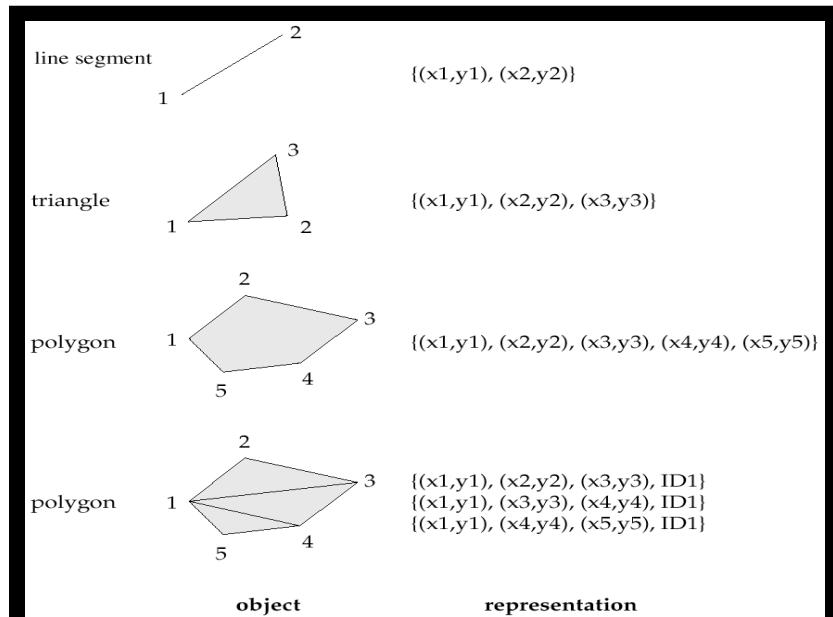
October 21st

Indexing for content based retrieval

- Content based retrieval from large collections of images (more in general of non-textual data) might require support of multi-dimensional index structures to speed-up retrieval.
- We can distinguish two different types of multi-dimensional indexes:
 - **Low dimensional indexes**, typically used in GIS and spatial database applications. They work with dimensions in the range of 2 - 4.
 - **High dimensional indexes**, typically used to index high dimensional feature vectors used as descriptors of images or image and video objects. They can work with dimensions in the range of 64 - 500

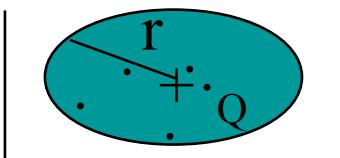
Low-dimensional Indexing

- Most common multidimensional data with low dimension are spatial data. They represent geometric entities: *points*, *line segments*, *triangles*, (in 2D) and *polyhedrons* (in 3D).
 - Geometric entities are usually represented in a normalized fashion:
 - Line segments
 - by the coordinates of their endpoints.
 - Curves
 - by partitioning the curve into a sequence of segments and creating a list of vertices in order
 - ...
 - Closed polygons
 - list of vertices in order, starting vertex is the same as the ending vertex,
 - dividing polygon into triangles and note the polygon identifier with each of its triangles.
 - ...

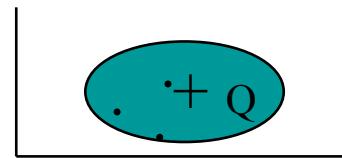


- For applications with spatial data, typical queries are concerned with spatial proximity. Different types of queries are:
 - Range Queries
 - E.g. : *Find all cities within 50 miles of Paris*
 - Query has associated region (location, boundary)
 - Answer includes overlapping or contained data regions
 - Nearest-Neighbor Queries
 - E.g. : *Find the 10 cities nearest to Paris*
 - Results must be ordered by proximity
 - Spatial Join Queries
 - E.g. : *Find all cities near a lake*
 - Join condition involves regions and proximity
- Most common applications of spatial data are:
 - Geographic Information Systems (GIS)
 - E.g., ArcInfo; OpenGIS Consortium
 - All classes of spatial queries and data are common
 - Computer-Aided Design/Manufacturing
 - Store spatial objects such as surface of airplane fuselage
 - Range queries and spatial join queries are common

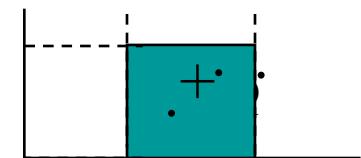
- For applications of content based retrieval in image collections the following query types should be supported:
 - **Vague queries**: Queries at the earlier stage can be very loose; e.g.: *Retrieve images containing textures similar to this sample.*
 - **K-nearest-neighbor-queries**: The user specifies the number of close matches to the given query point; e.g. *Retrieve 10 images containing textures directionally similar to this sample*
 - **Range queries**: An interval is given for each dimension of the feature space and all the records which fall inside this hypercube are retrieved.



r is large
vague query



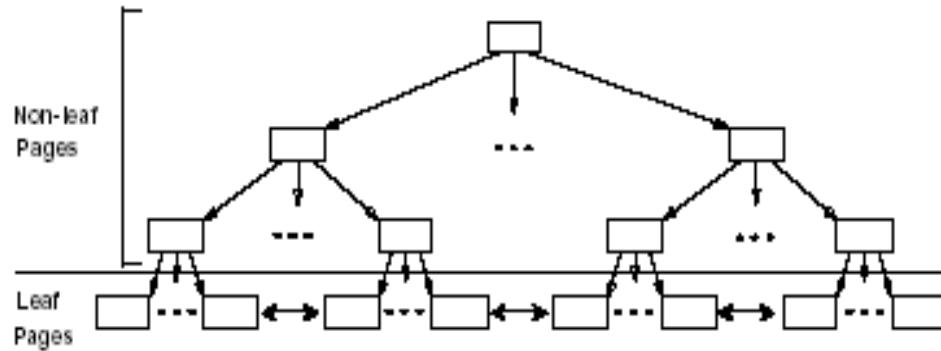
r is small
3-nearest neighbor query



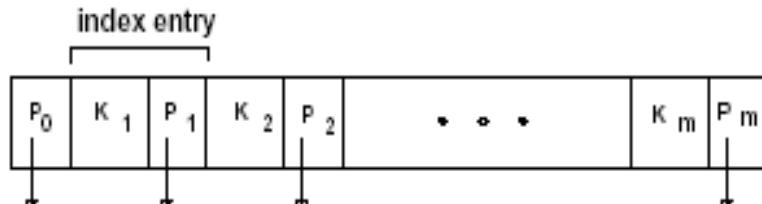
range query

Single-dimensional index

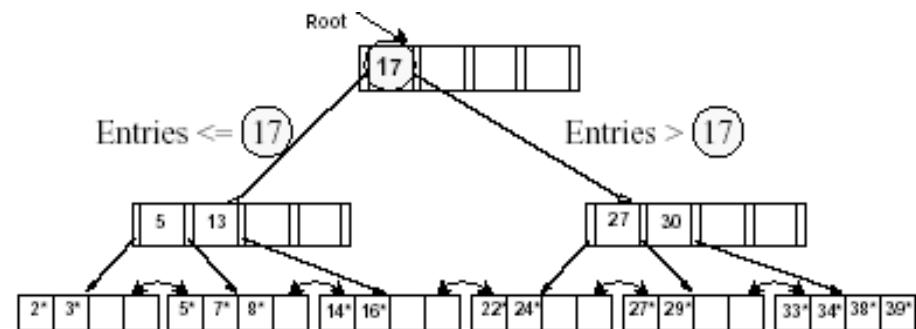
B+ Tree Index



- Leaf pages contain *data entries*, and are chained (prev & next)
- Non-leaf pages contain *index entries* and direct searches:



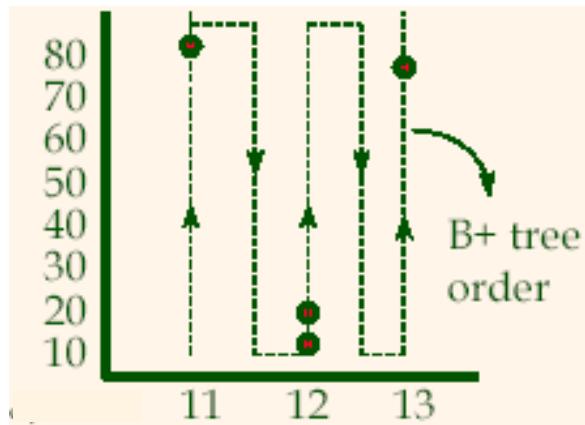
Example



- B+ Tree is a dynamic structure for unidimensional data
 - minimum 50% occupancy (except for root);
 - each node contains $d \leq m \leq 2d$ entries;
 - the parameter d is called the *order* of the tree;
 - typical order: 100;
 - typical fill-factor: 67%.
- B+ Tree supports equality and range-searches efficiently
- Inserts/deletes leave tree height-balanced
- B+ Tree have high fanout (this means depth rarely more than 3 or 4).
- Search cost: $O(\log(FN))$ ($F = \# \text{ entries/index pg}$, $N = \# \text{ leaf pgs}$)
 - search begins at root, and key comparisons direct it to a leaf;
 - find 28? 29? : All > 15 and < 30 .
- Insert/delete cost: $O(\log(FN))$
 - find data entry in leaf, then change it;
 - need to adjust parent sometimes;
 - and change sometimes bubbles up the tree.

- B+ Trees can be used for spatial data: when we create a composite search key e.g. an index on $\langle \text{feature1}, \text{feature2} \rangle$, we effectively linearize the 2-dimensional space since we sort entries first by feature1 and then by feature2 .

Consider entries:
 $\langle 11, 80 \rangle, \langle 12, 10 \rangle$
 $\langle 12, 20 \rangle, \langle 13, 75 \rangle$

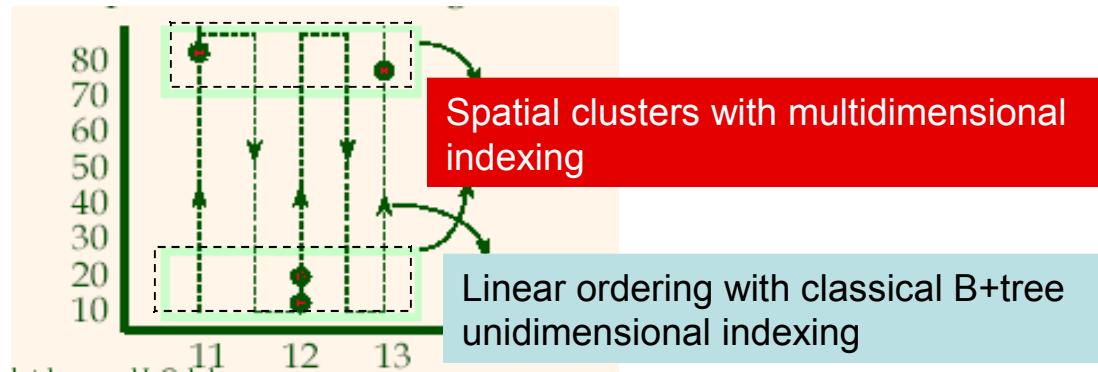


Multi-dimensional indexes

- A multidimensional index clusters entries so as to exploit “nearness” in multidimensional space.
- The basic motivation for multi-dimensional indexing is that for efficient content-based retrieval it is necessary to cluster entities so as to exploit *nearness* in multidimensional space.

Consider the entries:

<11, 80>, <12, 10>
<12, 20>, <13, 75>



- Most used multidimensional index structures:
 - k-d Trees
 - Point Quadtrees
 - R, R*, R+ Trees
 - SS, SR trees
 - M Trees
- ← For low/high-dimensional indexes
- ← For very high-dimensional indexes

Low-dimensional indexes

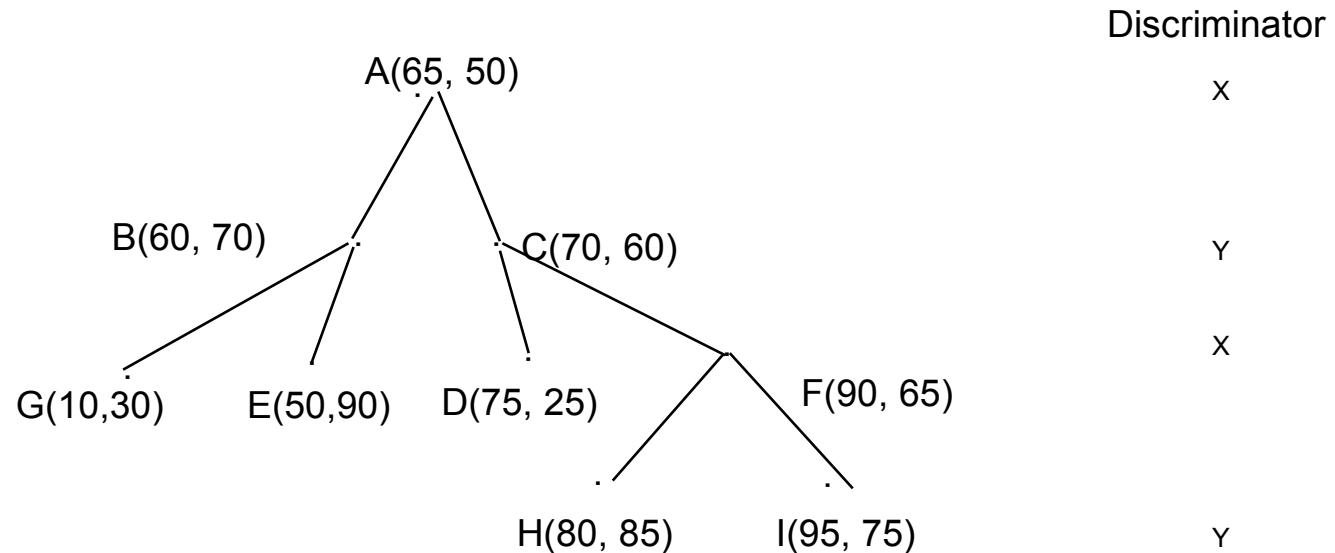
k-d Tree index

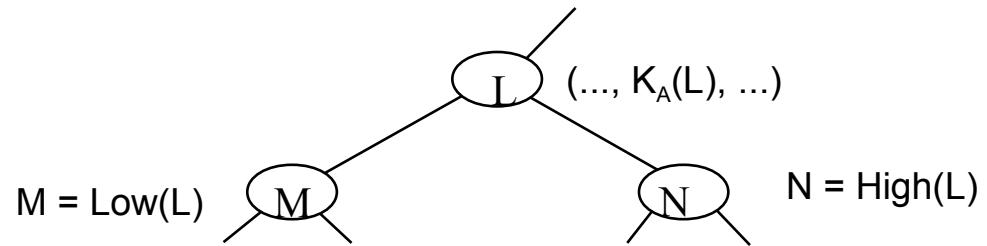
- The k-d Tree index (Bentley 1975) is a multidimensional binary search tree.
- Each node consists of a “record” and two pointers. The pointers are either null or point to another node.
- Nodes have levels and each level of the tree discriminates for one attribute
 - choose one dimension for partitioning at the root level of the tree
 - choose another dimension for partitioning in nodes at the next level and so on cycling through the dimensions.

Example

Input Sequence

A = (65, 50)
B = (60, 70)
C = (70, 60)
D = (75, 25)
E = (50, 90)
F = (90, 65)
G = (10, 30)
H = (80, 85)
I = (95, 75)





$\text{Disc}(L)$: The discriminator at L's level
 $K_A(L)$: The A-attribute value of L
 $\text{Low}(L)$: The left child of L
 $\text{High}(L)$: The right child of L

- Search for $P(K_1, \dots, K_n)$:

$Q := \text{Root};$

While NOT DONE DO the following:

if $K_i(P) = K_i(Q)$ for $i = 1, \dots, n$ then we have located the node and we are DONE

Otherwise

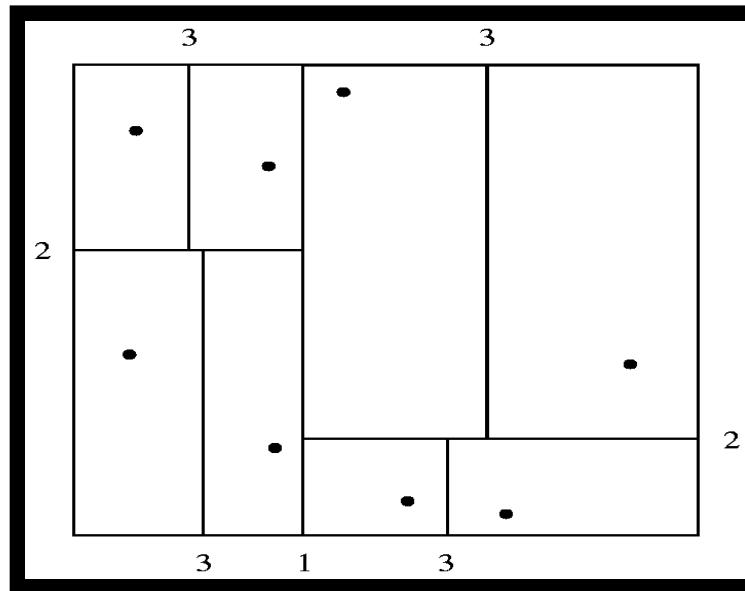
if $A = \text{Disc}(Q)$ and $K_A(P) < K_A(Q)$ then $Q := \text{Low}(Q)$

else $Q := \text{High}(Q)$

- Performance: $O(\log N)$, where N is the number of records

2-d Tree index

- 2-d Trees can be used for indexing point data in spatial databases.
- Division of space with 2-d Trees:
 - each line in the figure corresponds to a node in the k-d tree
 - In each node, approximately half of the points stored in the sub-tree fall on one side and half on the other.
 - Partitioning stops when a node has less than a given maximum number of points.



The numbering of the lines indicates the level of the tree at which the node appears.
The maximum number of points in a leaf node is set to 1.

- 2-d Trees are effective index structures for *range queries*.

Example of range query: *search those nodes of coordinates xy whose distance from vector (a,b) is less than d*

- The minimum cannot be less than $a-d, b-d$
- The maximum cannot be greater than $a+d, b+d$

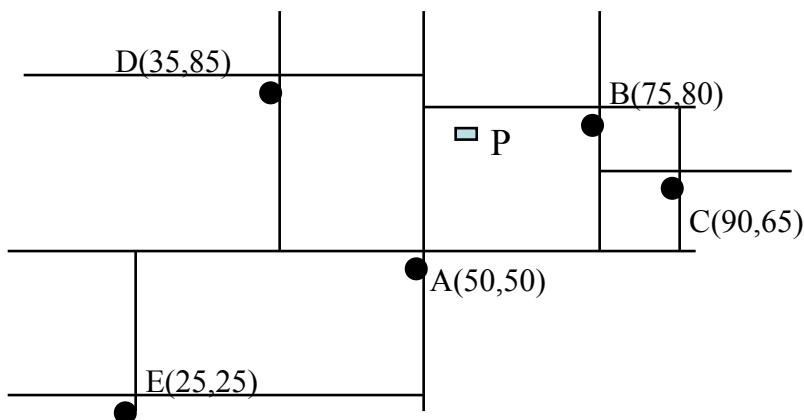
Therefore at each node x,y values must be compared with values stored in the node to decide which subtree has to be discarded

k-d Quadtree index

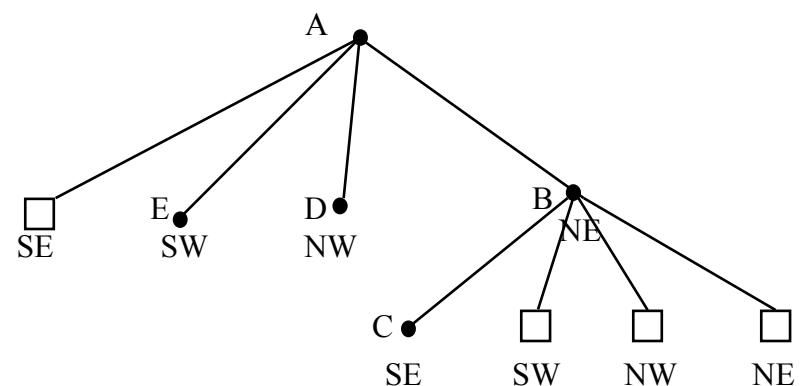
- In a k-dimensional Quadtree each node partitions the object space into quadrants. The partitioning is performed along all search dimensions and is data dependent, like k-d Trees.

Example of Quadtree for 2D points:

Partitioning of the space



The Quadtree

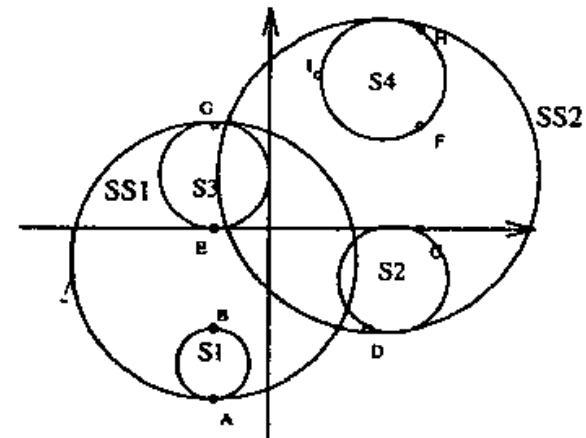
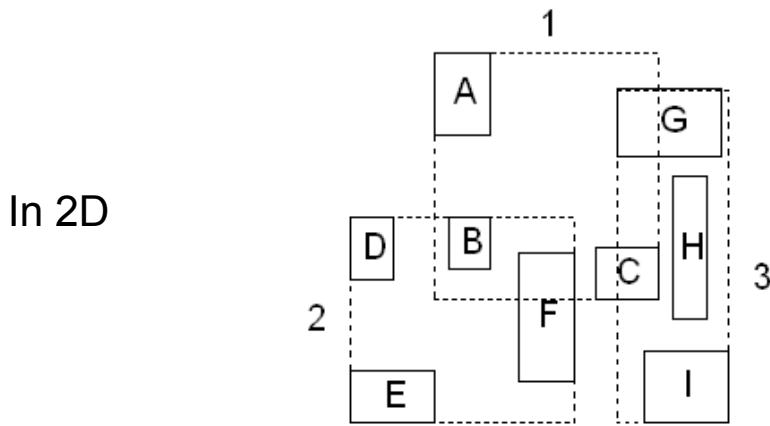


To insert P(55, 75):

- Since $X_A < X_p$ and $Y_A < Y_p$ go to NE (i.e., B).
- Since $X_p > X_p$ and $Y_p > Y_p$ go to SW, which in this case is null.

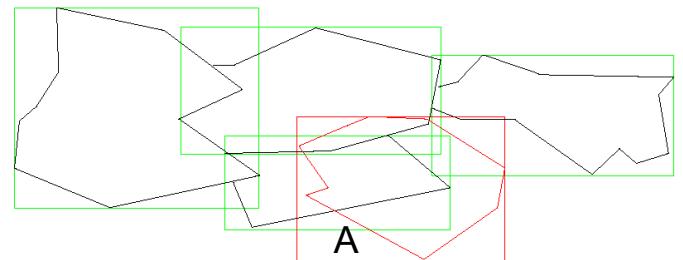
High-dimensional Indexes

- Image data are in two forms:
 - n-dimensional points (a feature vector describing the image or image objects)
 - n-dimensional objects with some spatial extension (for the evaluation of spatial relations)
- For image data high/very high dimensions indexes are needed. Typically these indexes should perform partitioning of the embedding space according to Minimum Bounding Regions (MBR) or Minimum Bounding Spheres (MBS) in the n-dimensional space. MBR and MBS refer to the smallest rectangle or circle respectively that encloses the n-dimensional entity (a region or a feature vector in n-d).
- As the number of dimensions increases the performance tends to degrade and most indexing structures become inefficient for certain types of queries

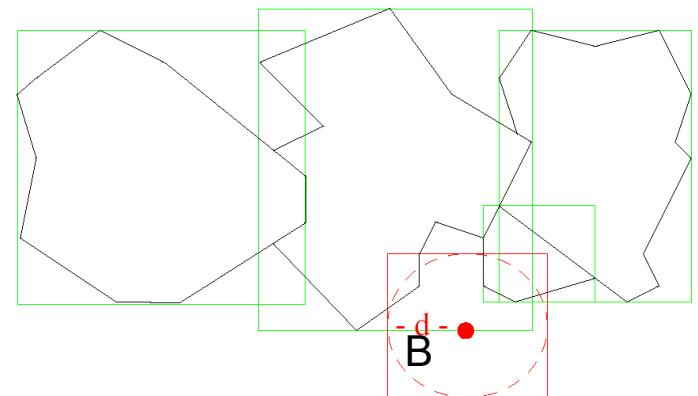


- Rectangles and circles are more difficult than points since do not fall into a single cell of a bucket partition. Therefore with these index structures it is necessary to manage overlapping of bucket regions.
- These index structures offer support also for new types queries:

- **Adjacency query**: find regions adjacent to A .

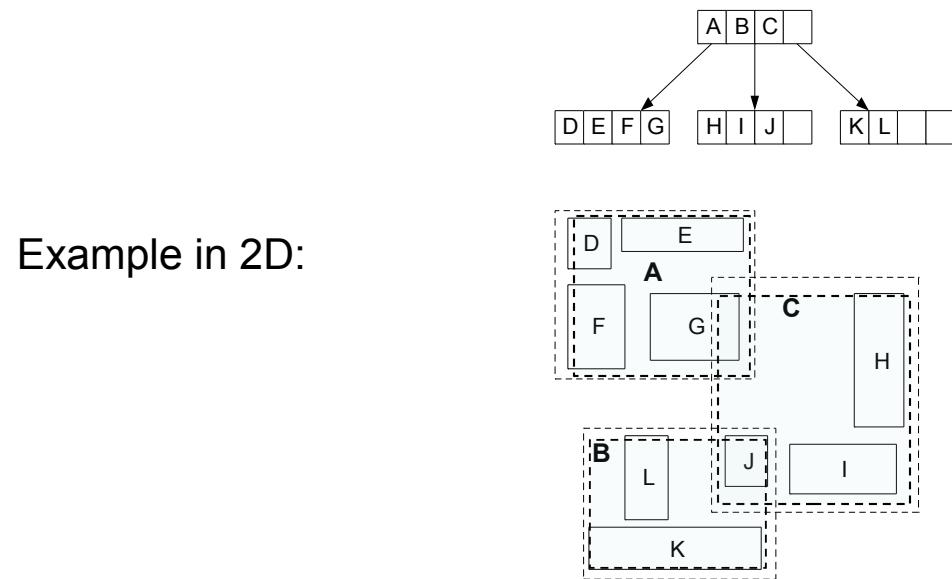


- **Distance qualified query**: find regions within distance d from B .

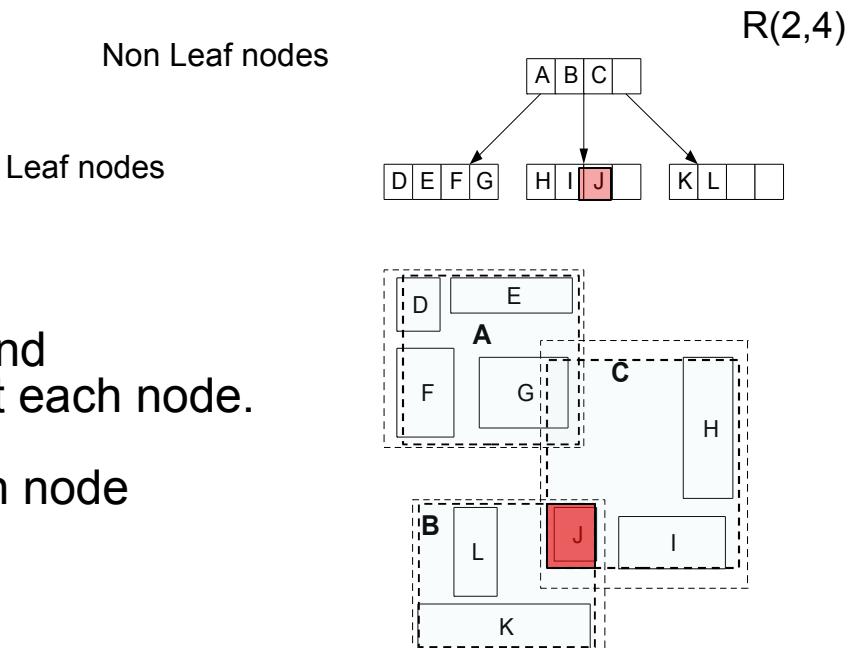


R Tree index

- The R Tree (Guttmann 1984) is a tree structured index that remains balanced on inserts and deletes. R Trees have been designed for indexing sets of rectangles and other polygons. Each key stored in a leaf entry is intuitively a box, or *collection of intervals*, with one interval per dimension.
- R Trees partition the space into rectangles, without requiring the rectangles to be disjoint. Enclosing rectangular regions are drawn with sides coinciding with the sides of the enclosed rectangles
- R Tree is supported in many modern database systems, along with variants like R⁺ Trees and R* Trees.



- Root and intermediate nodes correspond to the smallest rectangle that encloses its child nodes. Leaf nodes contain pointers to the actual objects:
 - Non-leaf entry = < *n-dim box, ptr to child node* > (box covers all boxes in child node (subtree))
 - Leaf entry = < *n-dimensional box, rid* > (box is the tightest bounding box for a data object)
- All leaf nodes appear at the same level (same distance from root).
- A rectangle may be spatially contained in several nodes (see rectangle J in the example), yet it can be associated with only one node.



- The **R Tree order (n,M)** is the minimum and maximum number of rectangles stored at each node.
- An R Tree of order (n,M) contains at each node between $n \leq M/2$ and M entries

- Search for objects overlapping box Q

Start at root.

- If current node is non-leaf, for each entry $\langle E, \text{ptr} \rangle$,
 - if box E overlaps Q, search subtree identified by ptr.
- If current node is leaf, for each entry $\langle E, \text{rid} \rangle$,
 - if E overlaps Q, rid identifies an object that might overlap Q.
- May have to search several subtrees at each node

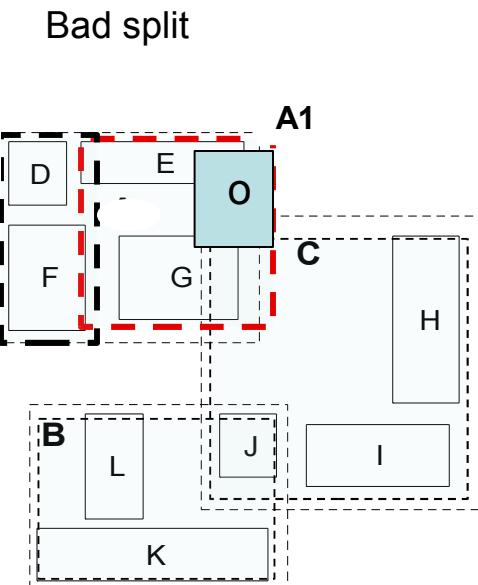
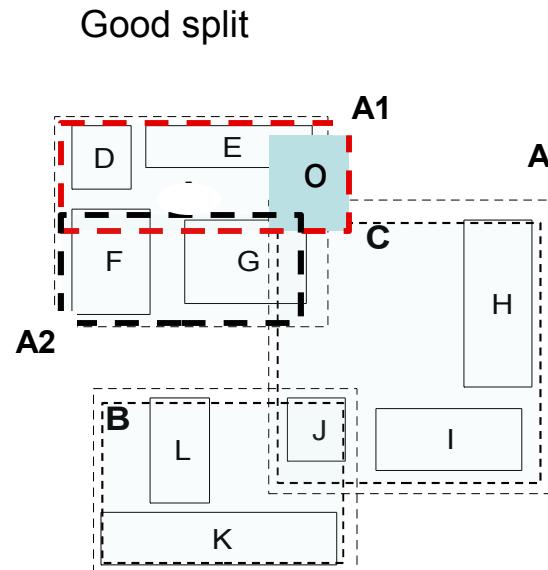
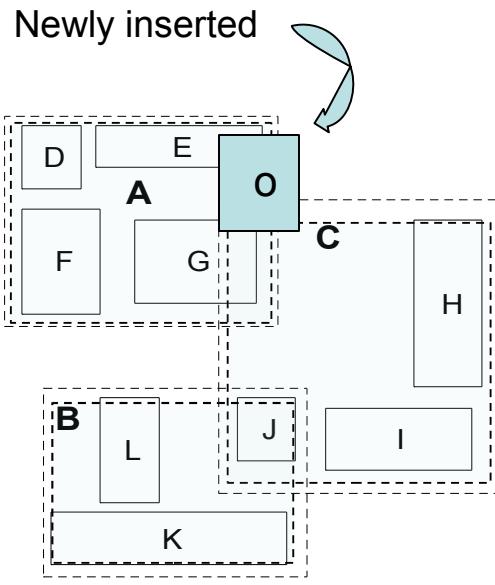
- Delete

- Delete consists of searching for the entry to be deleted, removing it, and if the node becomes under-full, deleting the node and then re-inserting the remaining entries.
- Remaining nodes are not merged with adjacent nodes as in B Tree.
There is no concept of adjacency in an R Tree.

- **Insertion**

- A new object is added to the appropriate leaf node.
- If insertion causes the leaf node to overflow, the node must be split, and the records distributed in the two leaf nodes. Goal is to reduce likelihood of both L1 and L2 being searched on subsequent queries:
 1. Minimizing the total area of the covering rectangles
 2. Minimizing the area common to the covering rectangles
- Splits are propagated up the tree (similar to B tree).

- Start at root and go down to best fit leaf L.
- Go to child whose box needs least enlargement to cover O
- Resolve ties by going to smallest area child
- If best fit leaf L has space, (A in ex) insert entry and stop. Otherwise, split L into L1 and L2 (A1, A2 in ex)
- Adjust entry for L in its parent so that the box now covers only L1.
- Add an entry in the parent node of L for L2. This could cause the parent node to recursively split.



Considerations on R Trees

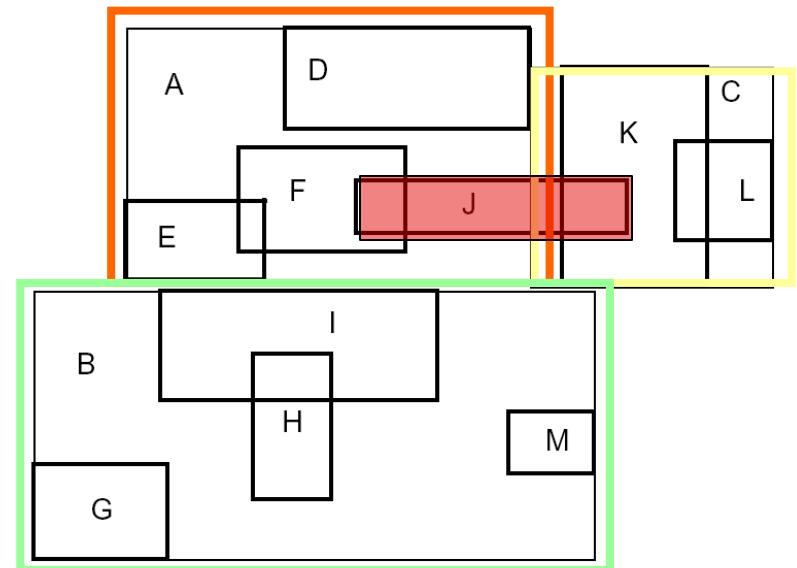
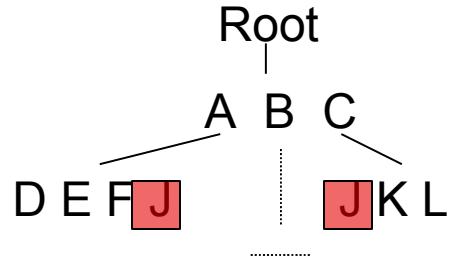
- For search
 - advantage: each spatial object (or key) is in a single bucket
 - disadvantage: multiple search paths due to overlapping bucket regions.
 - can improve search performance by using a convex polygon to approximate query shape (instead of a bounding box) and testing for polygon-box intersection.
- Generalization for higher dimension is straightforward, although R Trees work well only for relatively small n

R* Tree index

- The R* Tree is a variant of R Tree that uses the concept of forced reinserts to reduce overlap in tree nodes. When a node overflows, instead of splitting:
 - Remove some (say, 30% of the) entries and reinsert them into the tree.
 - Could result in all reinserted entries fitting on some existing pages, avoiding a split.
- R* Trees also use a different heuristic, minimizing box perimeters rather than box areas during insertion.

R+ Tree index

- R+ trees (Sellis, Rossopoulos & Faloutsos 87) are an alternative to R Trees that avoids overlap of enclosing rectangles by inserting an object into multiple leaves if necessary. One rectangle is associated with all the enclosing rectangles that it intersects.
- Data rectangles cut into several pieces, if necessary. The same rectangle can be reached from multiple paths starting from the root. Searches now take a single path to a leaf, at cost of redundancy.

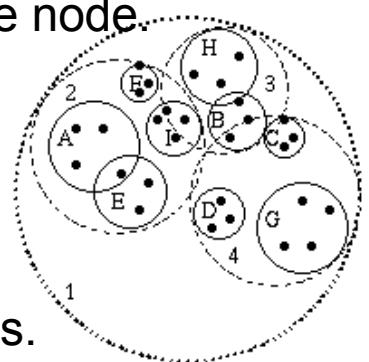


Very High-dimensional indexes

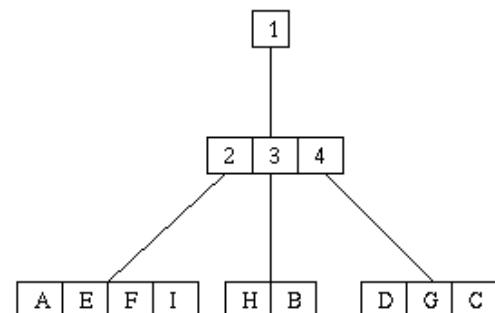
- Typically, high-dimensional datasets are collections of points, not regions and are very sparse.
 - E.g. feature vectors in multimedia applications.
 - R-tree becomes worse than sequential scan for most datasets with more than a dozen dimensions.
- As dimensionality increases, contrast (ratio of distances between nearest and farthest points) usually decreases;
 - “nearest neighbor” is not any more meaningful.
 - In any given data set, empirically test contrast.

SS Tree index

- The SS Tree index uses minimum bounding spheres (MBSs) to represent objects instead of MBRs. SS Tree index divides points into short-diameter regions, while R Tree divides points into small-volume regions.
- While $2 \times d$ real numbers are used to represent MBRs, only $d+1$ real numbers are used for MBSs (one for the sphere center and one for its radius). The space saving allows more entries to be fit in a tree node.

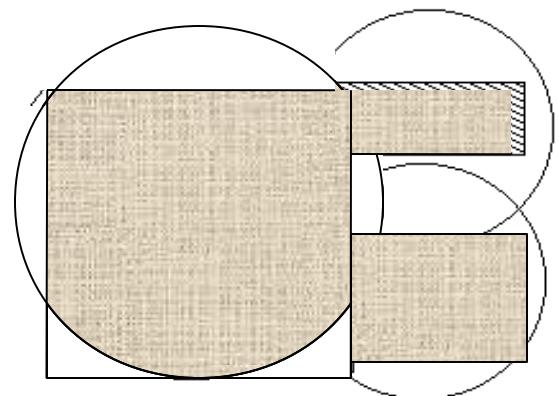


- In the SS Tree, points are divided into isotropic neighborhoods. The center of a sphere is the centroid of underlying points.



SR Tree index

- SR Tree index are based on the consideration that minimum bounding spheres occupy much larger volume than bounding rectangles with high dimensional data on average, and this reduces search efficiency.
- According to this in the SR Tree index, a region is specified by the intersection of a bounding sphere and a bounding rectangle.



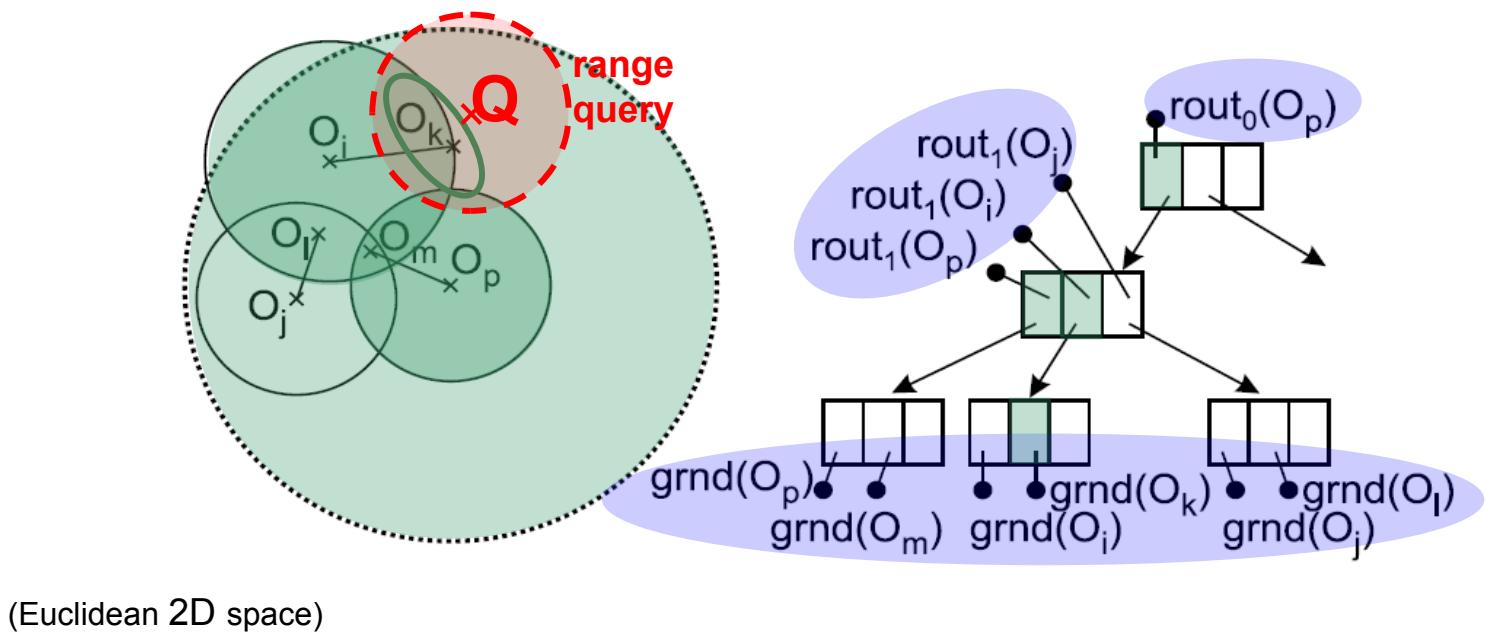
- The SR Tree index combines the advantages of both spherical and rectangular regions. In fact:
 - the average volume of bounding rectangles is much smaller than that of bounding spheres;
 - the average diameter of bounding rectangles is much longer than that of bounding spheres.

Distance Based Index Structures

- Distance based index structures also referred to as *metric trees* are proposed for applications where:
 - the distance computation between objects of the data domain are expensive (such as for high dimensional data)
 - the distance function is metric
- Metric trees only consider relative distances of objects to organize and partition the search space. The only requirement is that the distance function is metric, which allows the triangle inequality to be applied.
- Since metric spaces strictly include vector spaces, metric trees have more general applicability in content-based retrieval than multidimensional access methods.

M Tree index

- M Tree is a balanced paged metric tree like e.g. B⁺ Tree and R Tree, designed to act as a dynamic access method. Objects are collected in a hierarchical set of clusters:
 - The tree leaves are clusters of indexed objects O_i (ground objects)
 - Routing entries in the inner nodes represent hyper-spherical metric regions (O_i, r_{O_i}) , recursively bounding the object clusters in leaves
 - Each cluster has:
 - A reference (routing) object;
 - A radius providing an upper bound for the maximum distance between the reference object and any other object in the cluster.
- The compactness of metric regions' hierarchy in M-tree heavily depends on the order of new objects' insertions



- The triangle inequality allows to discard irrelevant M-tree branches (metric regions resp.) during query evaluation

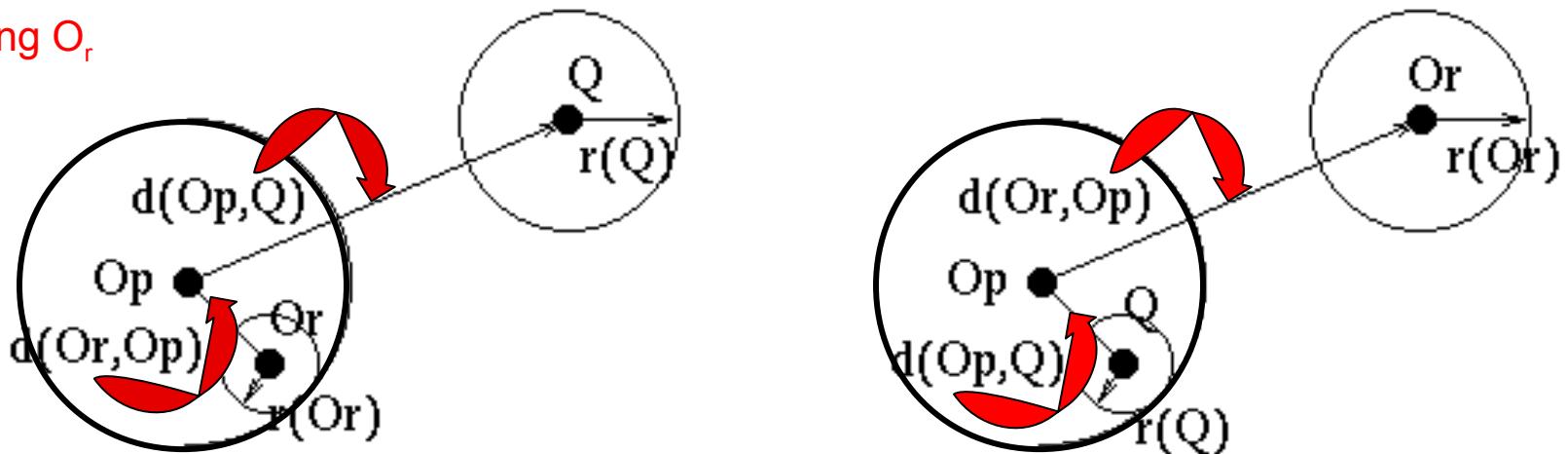
- To this end, an entry of the M Tree index in an internal node is constructed as follows:

$$entry(O_r) = [O_r, \text{ptr}(T(O_r)), r(O_r), d(O_r, P(O_r))]$$

- where:
 - O_r is the *routing object*
 - $\text{ptr}(T(O_r))$ is the pointer to the root of sub-tree $T(O_r)$ - the *covering tree* of O_r
 - $r(O_r)$ is the *covering radius* of O_r
 - $d(O_r, P(O_r))$ is the distance from the parent object

- Since the distance is metric, during search, the triangular inequality is used to prune clusters that are bound out of an assigned range from the query.

Pruning O_r

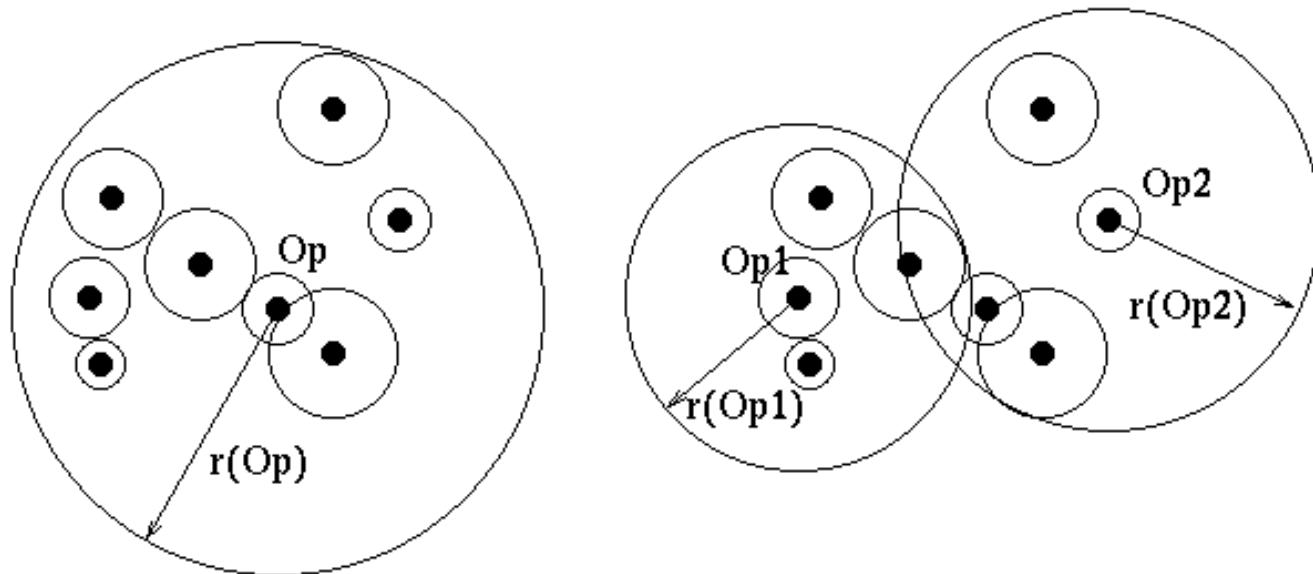


Given the range query $(Q, r(Q))$, the condition applied for pruning are as follows:

$T(O_r)$ can be safely pruned from the search if: $d(O_r, Q) > r(Q) + r(O_r)$

or if: $|d(O_p, Q) - d(O_r, O_p)| > r(Q) + r(O_r)$ $\text{or } d(O_r, Q) > r(Q) + r(O_r)$

- M-tree splitting



New infrastructures for Big Data: NoSQL Databases

VOLUME and VELOCITY

- A transaction represents the typical elementary unit of work of a Database Server
- A transaction identifies a unit of work performed by an application
- We also have VARIETY and VERACITY

Transactions

- The classical DBMSs (also distributed) are transactional systems: they provide a mechanism for the definition and execution of transactions
- In the execution of a transaction the ACID properties must be guaranteed
- New DBMS have been proposed that are not transactional systems

ACID

- Atomicity: A transaction is an indivisible unit of execution
- Consistency: the execution of a transaction must not violate the integrity constraints defined on the database
- Isolation: the execution of a transaction is not affected by the execution of other concurrent transactions
- Persistence (Durability): The effects of a successful transaction must be permanent

NoSQL databases

- It has been realized that it is not always necessary that a system for data management guarantees all transactional characteristics
- The non-transactional DBMSs are commonly called **NoSQL DBMSs**
- This really is not correct, because the fact that a system is relational (and uses the SQL language) and that it has transactional characteristics are **independent**

BIG DATA and the Cloud

DATA CLOUDS:

ON DEMAND STORAGE SERVICES, reliable, offered on the Internet with easy access to a virtually infinite number of storage resources, computing and network

CLASSIFICATION OF POSSIBLE METHODS OF STORAGE:

- **Centralized or distributed**, transactional, based on a traditional model (eg relational) remain the most widely used for traditional business applications (business information systems etc.).
- **Federated and multi-databases**, generally for companies and organizations that are associated and share their data on the Internet
- **Cloud databases**, to support BIG DATA by means of load sharing and data partitioning

NoSQL databases

- Provide flexible schemas
- The updates are performed asynchronously (no explicit support for concurrency)
- Potential inconsistencies in the data must be solved directly by users
- Scalability: no joins, ease of clustering, no 2PhaseCommit
- Evolution to a “simpler” schema: key/value-based, semi/non-structured
- Object-oriented friendly
- Caching easier (often embedded)
- Easily evolved to live replicas and node addition, made possible by the simplicity of repartitioning of data
- DO NOT support all the ACID properties

DATA MODEL

3/4 categories:

- Key–Value
- Document–based
- Column–family
- Graph-based

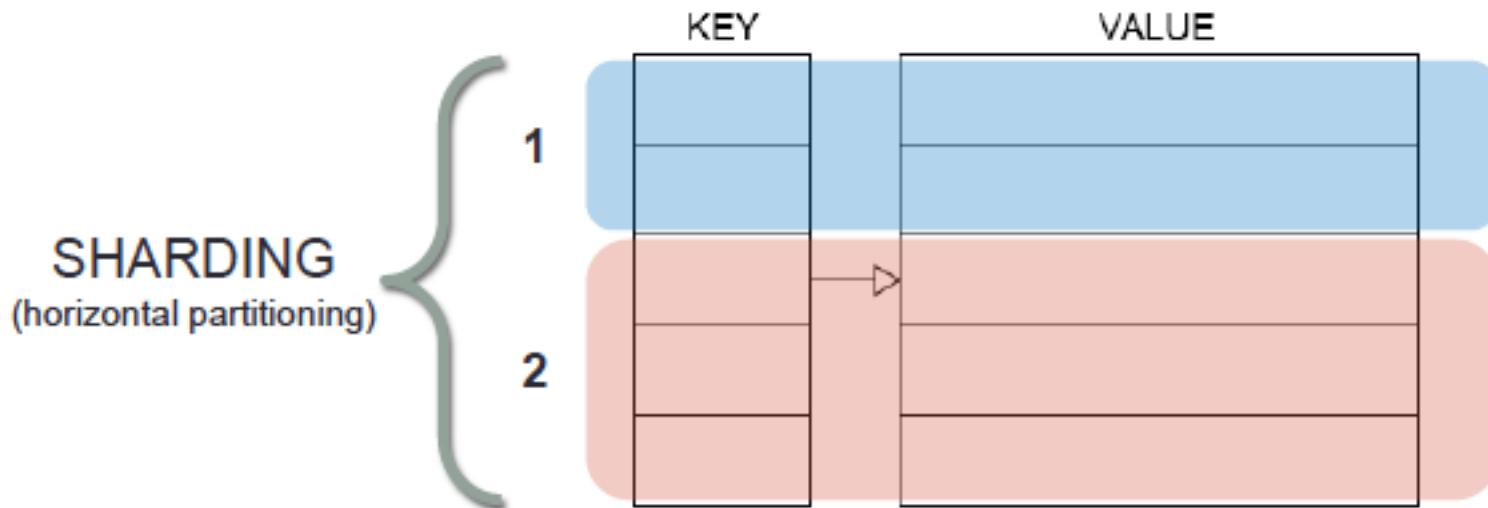
... Graphs not treated here, a separate evolutionary path from the other categories. They rely heavily on **modeling relationships**

Key -Value

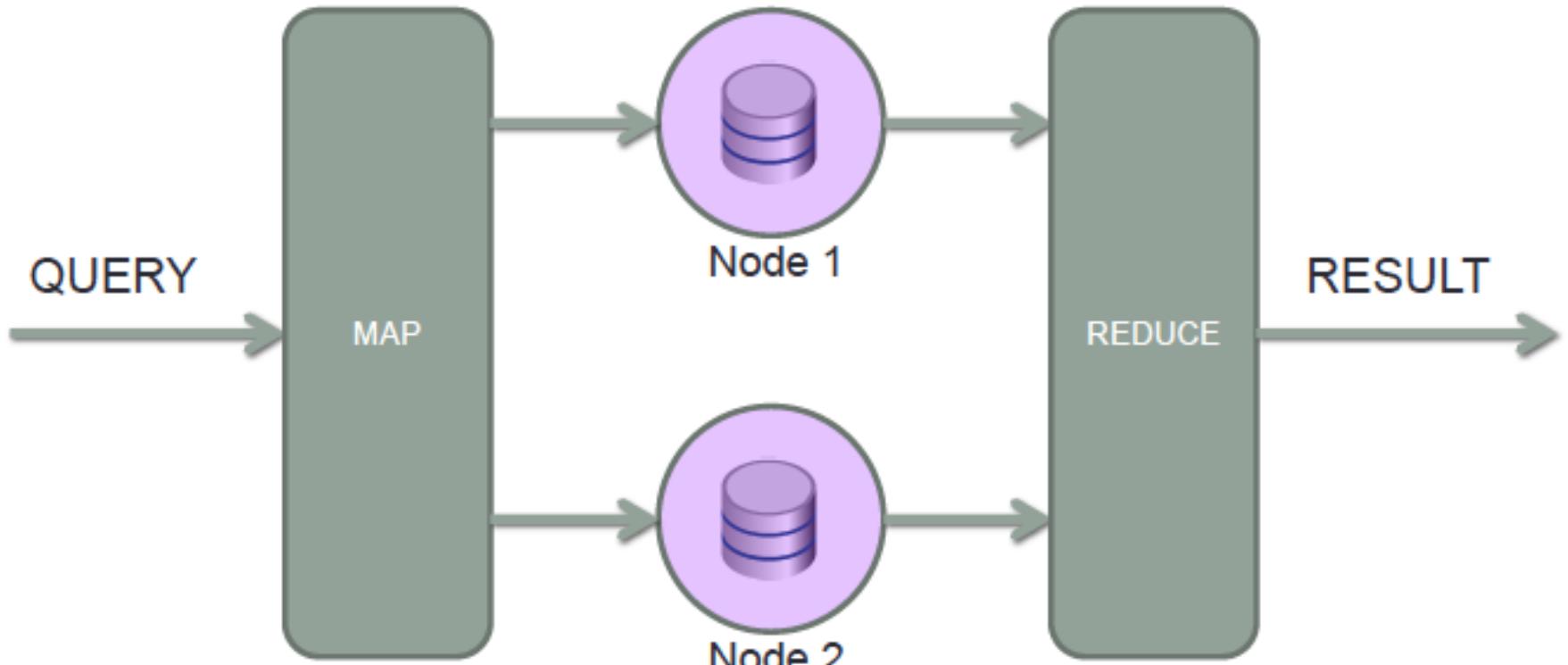
- Classical reference model of NoSQL systems
- Key: single or compound
- Value: blob, " opaque"
- Querying = find by key
- No schema (a dictionary)
- Standard APIs: get / put / delete

Key-Value: Scaling on multiple nodes

- Joins limit scalability
- No relationships in the database → easier to scale!
- Decoupled and denormalized entities are ‘self-contained’
- We can move them to different machines without having to worry about “neighborhood”!
- Sharding (horizontal partitioning)



Map Reduce



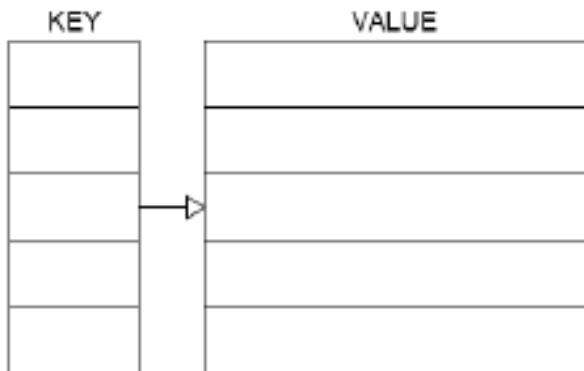
@Oscar Locatelli

Map-reduce paradigm

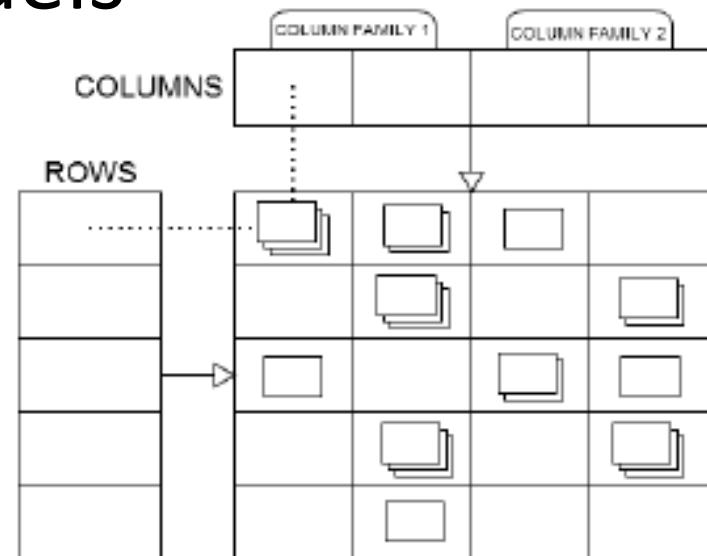
- Apache Hadoop was the first high-impact proposal
 - Hadoop 1.0 only had HDFS and MapReduce
- Hadoop 2.0 is more flexible
 - YARN, for resource description and management
 - Pig, Hive, Spark, Kafka (message broker), Sqoop (from DBMSs to Hadoop)
- Apache Spark is currently receiving a lot of attention
 - MapReduce stores in mass memory intermediate results, with high costs when executing multiple steps
 - Spark manages a direct transfer from one step to the next, similar to pipelines in DBMS query plans
- Several components. Among them:
 - MLlib (machine learning)
 - SparkSQL

Further Models

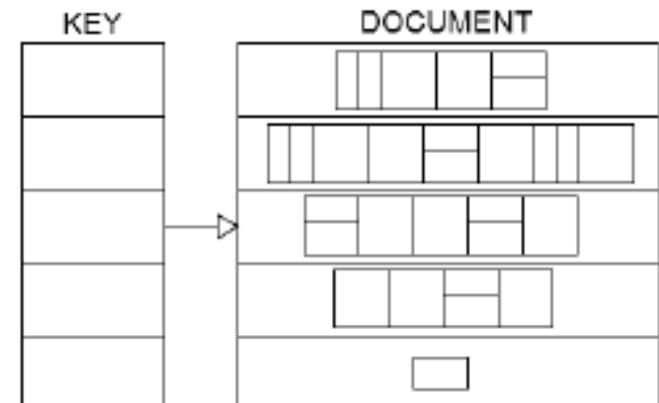
Key-Value



Column oriented



Document based



Column -Family 1/2

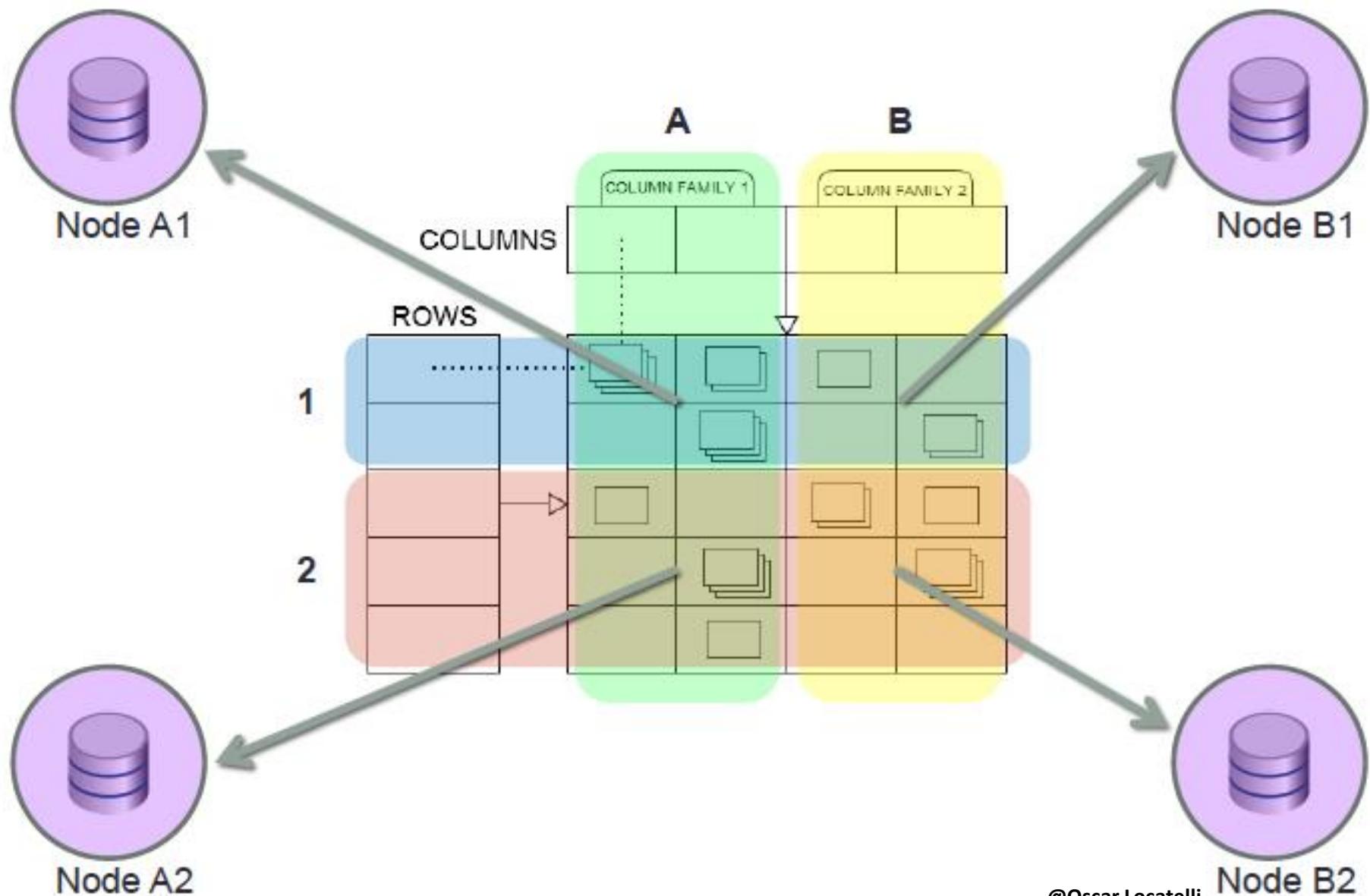
- Key is a triple row / column / timestamp
- Value: "opaque"
- Strongly oriented to Big Data, maximum scalability, the data is partitioned horizontally (sharding) and vertically (different columns on different machines) based on the keys of the row and column
- Ability to query: get or filter only on row and column keys (indexed). Supports projection
- Sorting is automatic, based on the names of rows and columns, so schema design is very important
- Standard APIs: get / put / scan / delete

Column–Family 2/2

Semi-structured diagram:

- Row: columns indexed within each row by a row-key
- Column-family: a set of columns, normally similar in structure to optimize compaction.
- Columns in the same column family will be “close”: to be determined at design time. Equivalent to tables in the RDBMS, but are somehow “semi-structured” (empty column-families occupy 0 bytes)
- Columns: have a name, indexed (column-key) and may contain a value (multi-version) for each row. Can be dynamically added (empty column occupies 0 bytes)

Column-family: maximum scaling



Document-based

- Key
- Value: collection of documents whose structure is not fixed and not even comparable to XML or JSON, sometimes “nested”
- Several mechanisms of the RDBMS, such as indexes, views, triggers, transactions (but very different in semantics)
- Ability to query: very rich: filters (and also indexes) on the internal fields of the document
- Schema-free and less ‘sensitive’ to design than a column-based
- API: high-level, conceptual (like ORM)

The CAP theorem

A data management system shared over the network (cloud, networked shared-data system) can guarantee at most two of the following properties:

- **Consistency** (C): all nodes see the same data at the same time
- **Availability** (A): a guarantee that every request receives a response about whether it was successful or failed
- (tolerance to) **Partitions** (P): the system continues to operate despite arbitrary message loss or failure of part of the system

According to Brewer's 2012 article, "2 of 3" is misleading:

- because partitions are rare, there is little reason to forfeit C or A when the system is not partitioned.
- the choice between C and A can occur many times within the same system at very fine granularity; not only can subsystems make different choices, but the choice can change according to the operation or even the specific data or user involved.
- all three properties are more “fuzzy” than binary

The CAP theorem

- In ACID, the C means that a transaction preserves all the database constraints
- the C in CAP refers only to copy consistency, a strict subset of ACID consistency.

From this we can understand why in more traditional applications, such as banking or accounting, or bookings etc. these systems can be catastrophic

Interesting applications:

- data collected from sensors (append-only)
- In general, datasets that are seldom updated

SOME FAMOUS IMPLEMENTATIONS

- Amazon DynamoDB
 - Key-value
 - CAP : AP - guarantees Availability and Partition tolerance, relaxing Consistency
 - auto – sharding
 - P2P networks
 - It aims to eliminate the job of the database administrator
 - Project Voldemort, SimpleDB
- Google BigTable
 - Column- oriented, on the Google BigTable paper serves as the foundation to the NoSQL Column- based data –model
 - CAP: CP - if there is a network partition Availability is lost, but ‘strict’ consistency may be required
 - auto-sharding, conflict resolution manual, no P2P

SOME FAMOUS IMPLEMENTATIONS (II)

- Hypertable and Hbase
 - Implementations of BigTable (built on Google File System)
 - Both within Apache Hadoop (framework for distributed applications based on map-reduce)
 - Interface Thrift, REST and APIs for various languages
 - HBase extensible (coprocessors), Hypertable most powerful
- Cassandra
 - Free from the Apache Foundation, Unix-like and Windows
 - Super-Column family
 - CAP : AP consistency with configurable auto - sharding, automatic conflict resolution
 - Combines the P2P with the data-model of BigTable
 - Transactions lock-free
 - Key names only on rows and columns
 - Also on Hadoop

SOME FAMOUS IMPLEMENTATIONS (III)

- Redis
 - Originally developed by Salvatore Sanfilippo
 - Key-value model
 - Supports storage in mass-memory, but it operates in main memory
 - High performance and scalability, often used as a cache backend for Websites
 - Some limited support is offered for joins, but it has a big impact on performance, as it runs scripts on the server

SOME FAMOUS IMPLEMENTATIONS (IV)

- MongoDB
 - Embeddable only in a C++ process, with LGPL license
 - Document-based
 - CAP: CP
 - auto-sharding with configurable strategy
 - static and automatic Indices created synchronously with the write
 - No transactions but atomic operations, and patterns for creating 2-phase commit or other policies
 - Query Task executed in Map-Reduce or otherwise distributed systems
 - In-place update of document attributes
 - Optimized for write-heavy (updates on index update and maintain consistency)
 - APIs for various languages ORM-like
- It is the most widely used and known, excellent performance and excellent documentation
- But, it is not a DBMS

SOME FAMOUS IMPLEMENTATIONS (IV)

- CouchDB
 - Document
 - CAP: AP, Multi-Version Concurrency Control, strict consistency of master, slave where appropriate
 - View materialized on the first read and updated with map-reduce algorithm written in Javascript, projection, sort and calculations
 - Transactions lock free
 - Write-heavy, Read-heavy
 - CouchDB is also a WebServer, can do application hosting HTML5 + JavaScript that are treated as documents (then synchronized between multiple databases, easy load-balancing)
 - Couchbase, CouchDB offers Memcached + GeoIndex
 - CouchDB is the basis of the synchronization service Ubuntu One

Different products for different objectives

- Enterprise Big Data: maximum performance and scalability: Amazon DynamoDB, Google BigTable and all their derivatives (Voldemort, Cassandra, HBase, Hypertable)
- Document Management or RDBMS replacement (with extreme caution!!!): MongoDB (write heavy) RavenDB (read-heavy), BigCouch (both not discussed here)
- Scenarios with online-offline, mobile devices/laptops: CouchDB (Ubuntu One)

Other interesting solutions

- CockroachDB
 - Emphasizes reliability, scalability and distribution
 - It is serverless
 - It uses Raft, an evolution of Paxos commit
- Firebase
 - Developed by Google, well supported within GCP
 - It goes beyond classical DBMS tasks (it is presented as an app development platform: auth, log collection, ...)
 - Firestore is now the NoSQL DBMS component
- Supabase
 - Open source alternative to Firebase
 - It starts from PostgreSQL
 - It offers scalability (sharding) and sophisticated access control, adapting triggers to the management of sophisticated rules

Bibliography on NoSQL

BOOK :

Tamer Ötzu M., Valduriez P. – Principles of *Distributed Database Systems*: 3rd ed. - Springer, 2011

More references

- Abadi Daniel J. - *Data Management in the Cloud: Limitations and Opportunities* - IEEE Data Engineering Bulletin, Vol. 32 No. 1, March 2009
<http://sites.computer.org/debull/A09mar/A09MAR-CD.pdf#page=5>
- Dean J., Ghemawat S. – *MapReduce: A Flexible Data Processing Tool* - CACM, Vol.53, n. 1, pp. 72-77, 2010
- Foster I., Yong Zhao, Raicu I., Lu S - *Cloud Computing and Grid Computing 360-Degree Compared* - Grid Computing Environments Workshop 2008, pp. 1-10, 2008
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4738445>