Privacy and Utility in Database Anonymization

Siddharth Srivastava, Philipp Weis

December 2005

Outline

k-Anonymity

- Introduced in [Sweeney, 2002].
- Ensures that any individual cannot be distinguished within a group of at least k individuals.
- This is achieved by generalizing attribute values to ranges.
- Finding an optimal generalization is NP-hard [Meyerson and Williams, 2004].

Goals of the Project

- Evaluate existing algorithms for k-anonymization.
- Find ways to measure the utility and privacy of a k-anonymized database.
- How can the anonymization algorithm be adapted to an expected workload?
- What is a good choice of k?

Party Donations Database

- Donations to political commitees in 2005 from [Federal Election Commision].
- 277,274 tuples, 164,649 unique quasi-id combinations

State	ZIP	Employement	Party	Amount
ID	83605	RETIRED	REP	400
TX	75206	AMERICAN AIRLINES	DEM	1000
TX	76115	PIER ONE IMPORTS	DEM	250
NJ	08844	BUSINESS OWNER	DEM	2000
ID	83651	SELF-EMPLOYED	REP	200
CA	92692	ATTORNEY	REP	200

Health Database

- National Ambulatory Medical Care Survey from 1973 [National Center for Health Statistics].
- 29,102 tuples, 9,678 unique quasi-id combinations
- Quasi-Identifiers: month of birth, year of birth, gender, race (white, other), region (NC, NE, S, W)
- Sensitive attributes: symptoms, diagnosis, x-rayed, sent-to-therapy, family-planning-related

06	1900	F	W	NE	3220	C	0	Y105	1	0
06	1930	M	W	NE	4150	C	0	7287	0	0
06	1907	F	O	NW	3060	C	0	4920	1	0
06	1900	F	W	NE	3060	A	0	4920	0	0
06	1906	F	O	NC	9900	C	0	4129	1	0

Outline

Multidimensional k-Anonymity

[LeFevre et al., 2006] propose a clean and effective approximation algorithm that can also be adapted to an expected workload.

Algorithm 1: Group.anonymize()

```
1 while self.is_splittable() do
2 |   split_attrib ← self.find_split_attrib()
3 |   (group1, group2) ← self.split(split_attrib)
4 |   return group1.anonymize() ∪ group2.anonymize()
5 return self.generalize()
```

The tricky part is to come up with a good heuristic to find the attribute to split on.

Output of the Anonomization Algorithm

[FL, GU]	[96932, 99401]	PAXSON COMMUNICATIONS CORP	REP	2000
[FL, GU]	[96932, 99401]	PAXSON COMMUNICATIONS CORP	DEM	300
[FL, GU]	[96932, 99401]	PAXSON COMMUNICATIONS CORP	DEM	300
[FL, GU]	[96932, 99401]	PAXSON COMMUNICATIONS CORP	DEM	1000
[FL, GU]	[96932, 99401]	PAXSON COMMUNICATIONS CORP	REP	300
[FL, GU]	[96932, 99401]	PAXSON COMMUNICATIONS CORP	DEM	500
[FL, GU]	[96932, 99401]	PAXSON COMMUNICATIONS CORP	DEM	500
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	500
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	2000
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250
MA	01002	[AMHERST COLLEGE, BULKELY RICHARDSON]	DEM	250

Outline

Generic Metrics

We want to capture the idea of information loss in the anonymization.

- *Discernability*:

$$\sum_{\text{groups } g} |g|^2$$

- *Normalized Average Group Size*:

$$\left(\frac{\#\text{tuples}}{\#\text{groups}} \right) / k$$

Problems with these metrics:

- They are more of measures of accuracy of the k-anonymization.
- They equate small groups with small information loss - which is not true.
- They capture both utility and privacy!

Workload Specific Metrics

- Define a set Q of statistical queries that represent an expected workload.
- *Average Relative Error*:

$$\frac{\sum_{q \in Q} \frac{\Delta q(anonDB)}{q(origDB)}}{|Q|}$$

$\Delta q(anonDB)$ is the difference between maximum and minimum possible answers of q on the anonymized database

Outline

Heuristics for Finding a Good Split

- [LeFevre et al., 2006] suggest to choose the attribute "with the widest (normalized) range of values," but don't give any reasons for this.
- "It might be possible to choose a dimension based on knowledge of an anticipated workload."
- Our experiments show that the choice of split attributes is crucial to the usefulness of the data.

Attribute Dependency

- Trivial dependencies, like ZIP $\longrightarrow$ state.
- Partial dependencies: In the party donations database, the company name is likely to determine the state, and usually corresponds to a small number of zipcodes. Conversely, the state restricts the possible companies.
- It turns out that splitting on state is more useful.

How to Choose Heuristics: Philosophy

- Partial matches: biggest source of error in queries on anonymized data.
- As splitting proceeds, values of non-split attributes become more and more unsorted. (Shuffling)
- Queries on numeric attributes are often interval queries: merging shuffled values destroys information.

Choosing Heuristics

Scheme 1: Generic

Split on the attribute with most repeated values first

- Information loss is reduced as similar valued tuples are merged.
- Minimizes merges and reduces partial matches.
- What if query is on another attribute?
- Increases shuffling on attributes with distinct information; shuffling on this attribute might have been safer.

Scheme 2: Query Specific

- If queries are known, split on basis of a partial order:
- "Don't split on attribute A before splitting on attribute B".
- Extremely query-specific: No concern for other attributes or information loss.

Heuristics Used

Generic

- Split on attribute with most number of similar values.
- Split on attribute with least number of similar values.

Query Specific, or Ordering-based

- Always split in this order: $Company < State < Zip$

  Great on...

  Company queries!
- Always split in this order: $State < Company < Zip$

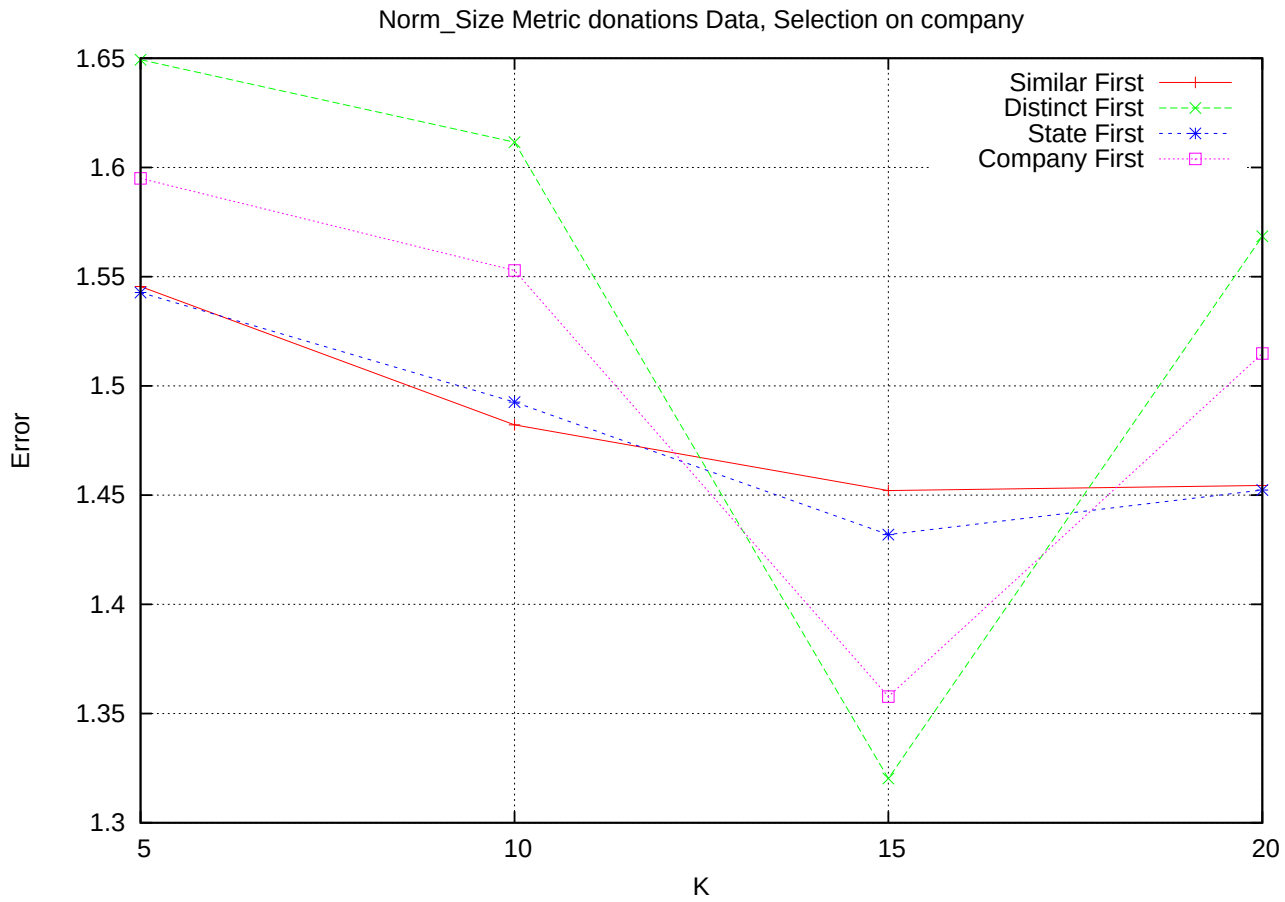
  Great on...

  State queries!

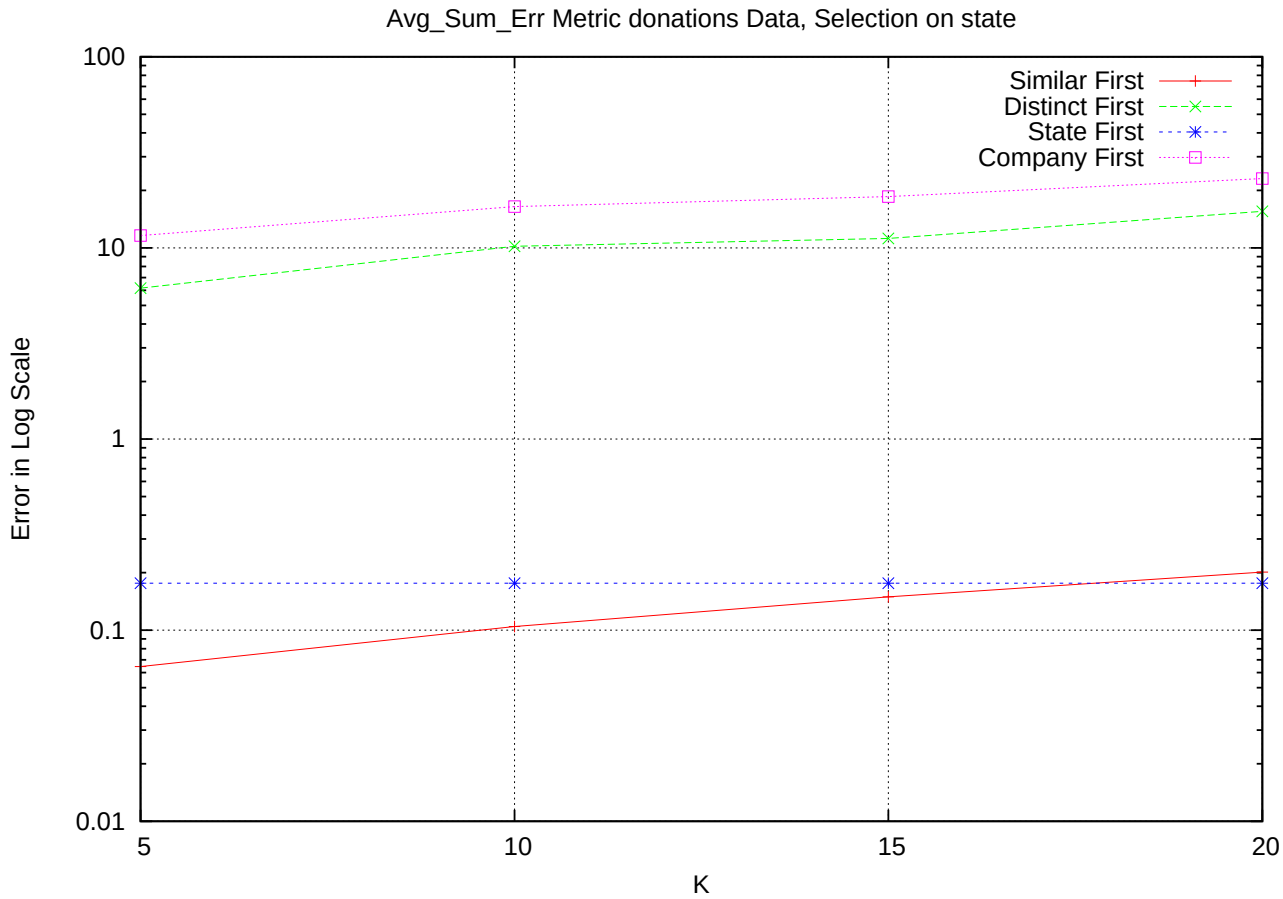
How do they compare on our set of metrics?

Outline

What the generic, normalized size heuristic says:

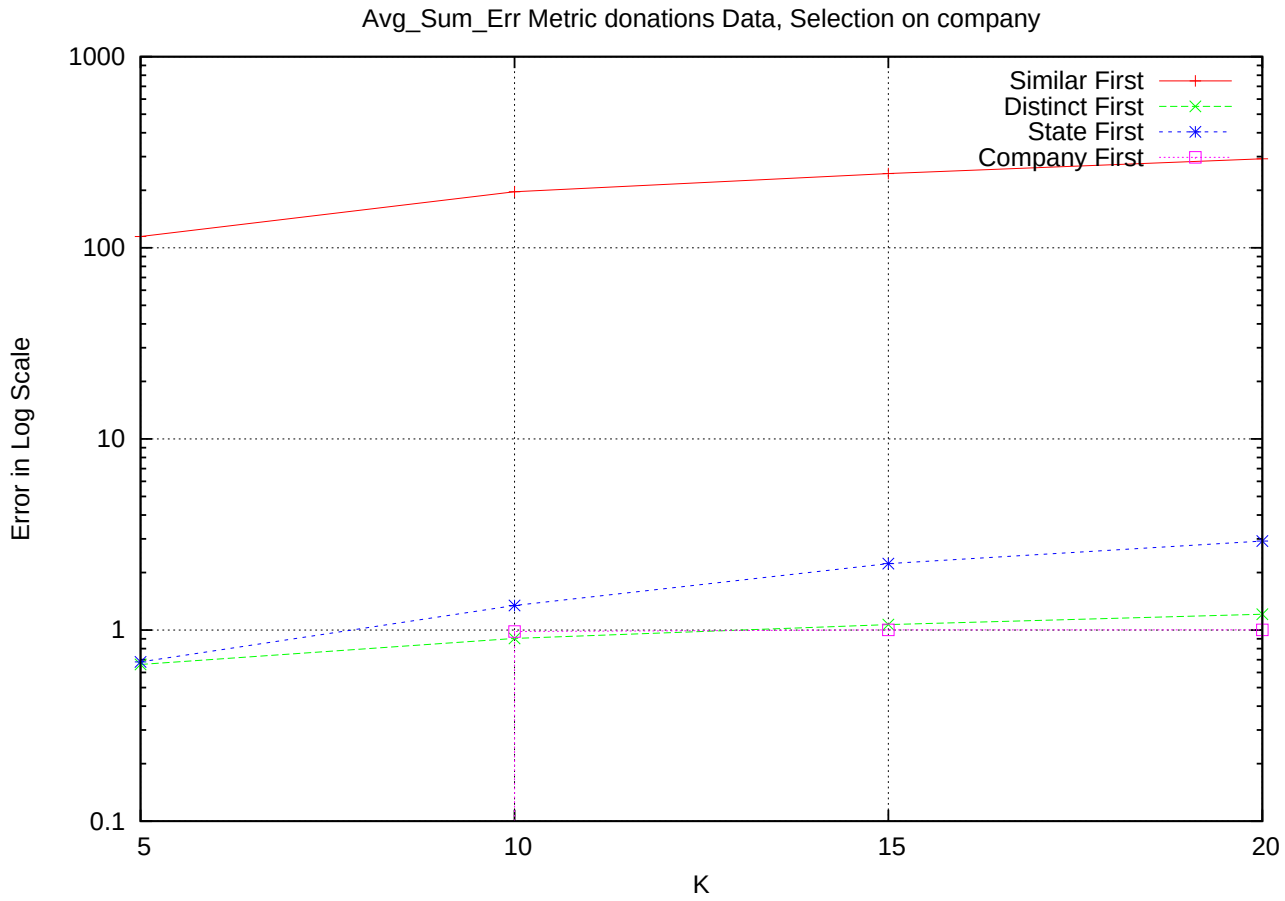


How does this compare with query-specific metrics?



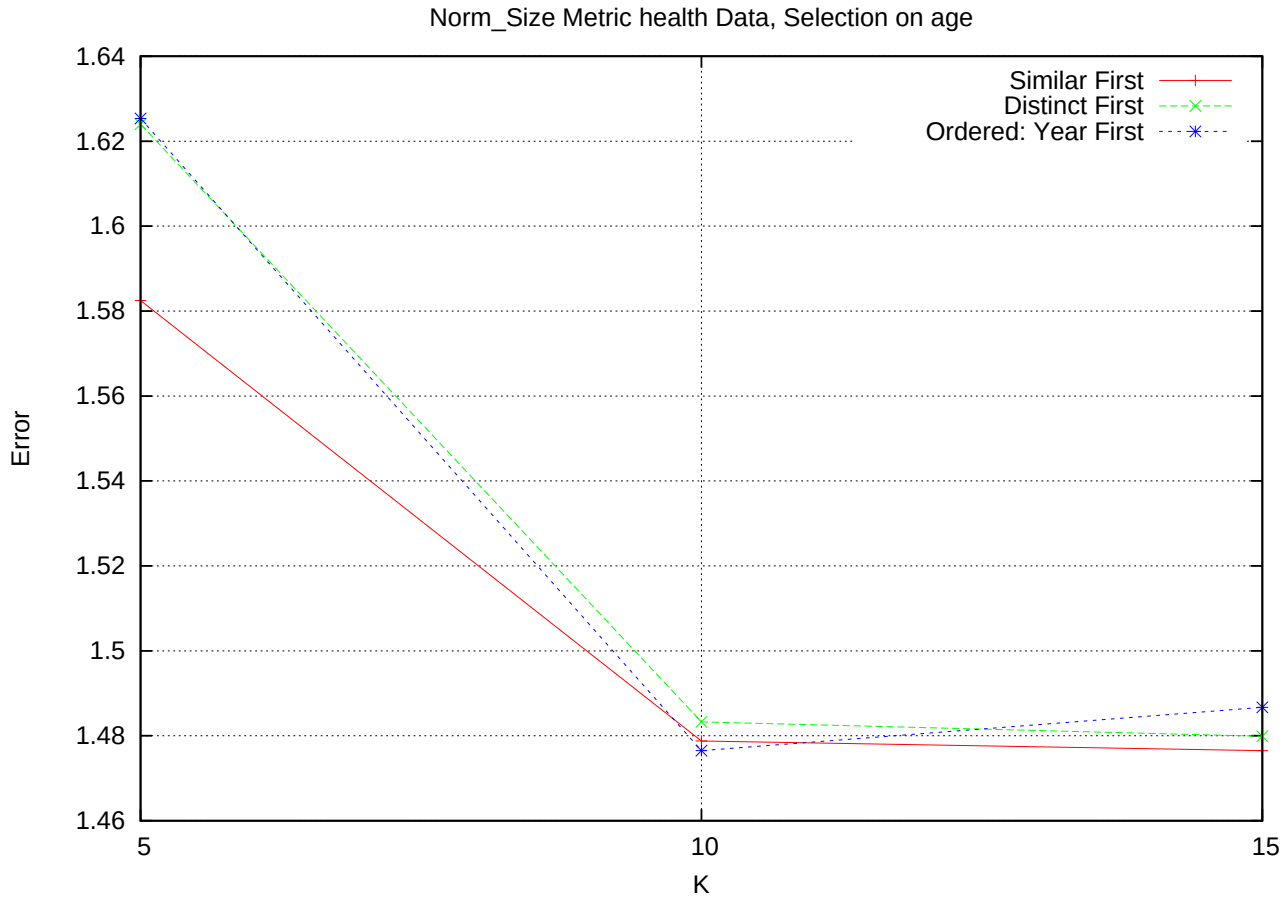
Splitting on the attribute with largest number of similar values, or on State first clearly benefits selections on State.

But this is disastrous for selections on Company:



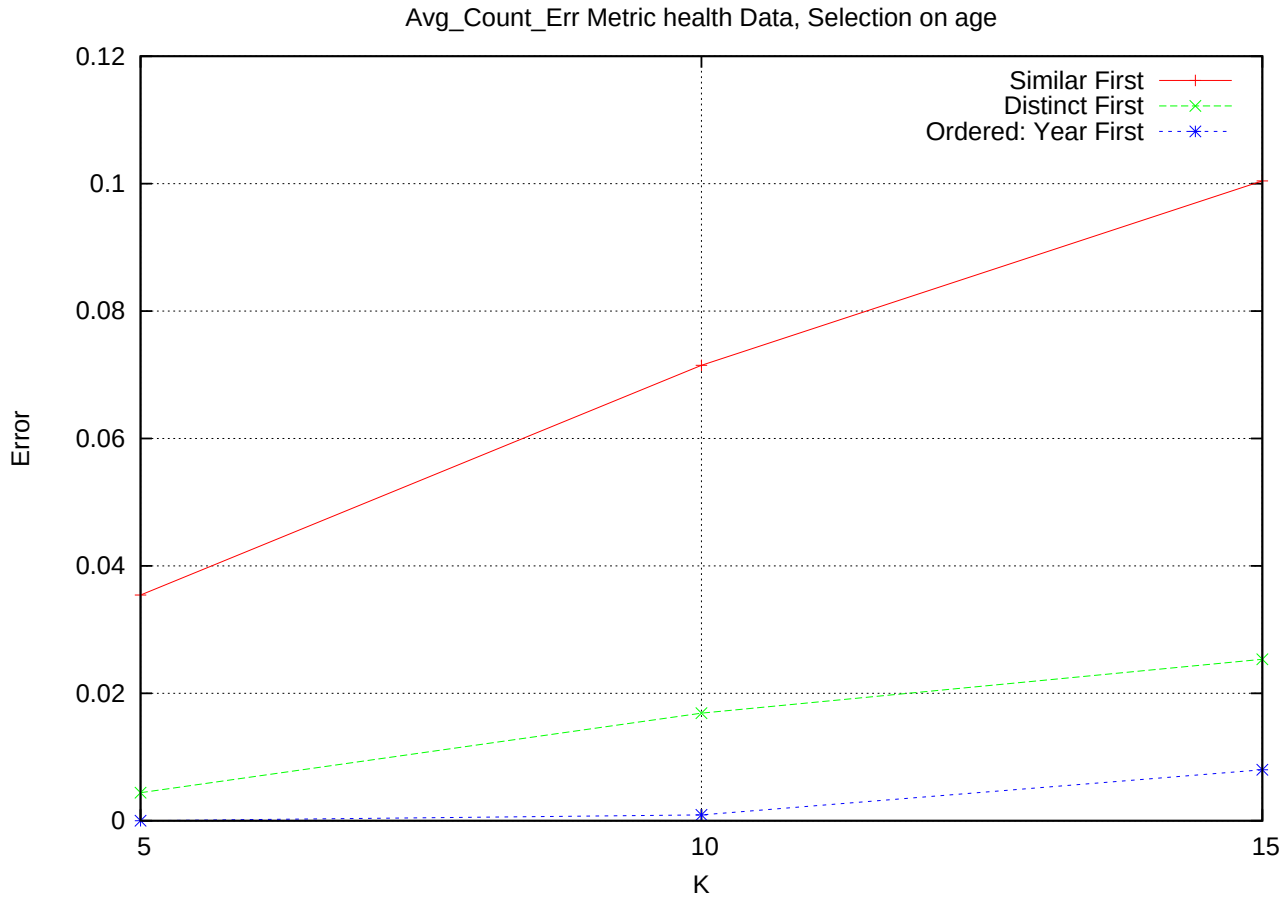
Similar observations hold for Count queries.

Measurement Results on the Health Database



Normalized group size metric not very useful.

Workload Metrics on the Health Database



Almost perfect results on Age queries!

Outline

Conclusions

- If we know what queries to expect, we can come up with good measures on the utility of an anonymization.
- Generic heuristics for the anonymization can produce really bad results for specific queries.
- On average, the generic heuristics showed better performance on the generic normalized-size metric, but performed poorly on the query-specific metrics.

References

Federal Election Commision. Campaign Finance Recipients and Data. URL `http://www.fec.gov/finance/disclosure/ftpdet.shtml`.

Kristen LeFevre, David DeWitt, and Raghu Ramakrishnan. Multidimensional *k*-Anonymity. *IEEE ICDE*, 2006.

Adam Meyerson and Ryan Williams. On the Complexity of Optimal *k*-Anonymity. *PODS*, 2004.

National Center for Health Statistics. Ambulatory Health Care Data. URL `http://www.cdc.gov/nchs/about/major/ahcd/ahcd1.htm`.

Latanya Sweeney. *k*-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10:557–570, 2002.