Achieving Privacy-Utility Trade-off in existing Software Systems

Saurabh Srivastava, Vinay P. Namboodiri, T.V. Prabhakar
Department of Computer Science & Engineering, IIT Kanpur, India

International Conference on Advanced Information Systems and Engineering
Cairo, Egypt

August 23-25, 2019

The Agenda for next 15 minutes !!

- Privacy vs Utility
  - Why it is difficult to achieve both?
  - How to choose a "sweet spot" on this "trade-off scale"?

The Agenda for next 15 minutes !!

- Privacy vs Utility
  - Why it is difficult to achieve both?
  - How to choose a "sweet spot" on this "trade-off scale"?
- The *Trade-off Model*
  - Can someone with no or very little understanding of data science make decisions about this trade-off?
  - What new "skills" would be required to do this analysis?

The Agenda for next 15 minutes !!

- Privacy vs Utility
  - Why it is difficult to achieve both?
  - How to choose a "sweet spot" on this "trade-off scale"?
- The *Trade-off Model*
  - Can someone with no or very little understanding of data science make decisions about this trade-off?
  - What new "skills" would be required to do this analysis?
- Engineering additions
  - Reducing the size of the problem space
  - Reducing the size of individual tasks

Privacy vs Utility

Motivation and understanding the problem

"Privacy" in applications using Data

- There is no "universally accepted" definition of exactly what "privacy" means
- Usually, *Privacy* is considered as the ability of an individual or an organisation to control what information about him or them gets exposed to the outside world
- Consequently, a "breach of privacy" is an event where some information about the individual or the organisation is "leaked" to someone that was not explicitly authorised
- Applications that use user data, need to make sure that user's privacy concerns are met

"Utility" in applications using Data

- Data is at the core of multiple activities in modern applications
- It is used to recommend products and services, customise content on social media, provide personalised discounts etc.
- The main idea about the *Utility* of data is extracting useful knowledge out of it, which can be applied for achieving business goals
- Applications that use user data, try to maximise the information that they can collect about their users, so that they can use it to provide better products and services

Achieving Privacy as well as Utility

- Utility is about "finding correlations in data"

Achieving Privacy as well as Utility

- Utility is about "finding correlations in data"
- Privacy is about "removing correlations in data"

Name	Roll Number	Department	Program	Income Range
Bob	1003	ME	BT	50K - 100K
Alice	1002	CSE	MS	>500K
John	1004	PHY	MT	100K - 350K
Mary	1005	CSE	PHD	50K - 100K
José	1006	MTH	BS	350 - 500K

Name	Roll Number	Department	Program	Incon
Bob	1003	ME	BT	50K - 100K
Alice	1002	CSE	MS	>500K
John	1004	PHY	MT	100K - 350K
Mary	1005	CSE	PHD	50K - 100K
José	1006	MTH	BS	350 - 500K

This data can be used to identify financially weaker students

Alice doesn't want this information to be public

Name	Roll Number	Department		ange
Bob	1003	ME		- 100K
Alice	1002	CSE	MS	>500K
John	1004	PHY	MT	100K - 350K
Mary	1005	CSE	PHD	50K - 100K
José	1006	MTH	BS	350 - 500K

Achieving Privacy as well as Utility

- Utility is about "finding correlations in data"
- Privacy is about "removing correlations in data"
- Ways to remove "correlations"
  - Anonymise data (Alice ⇒ P1, Bob ⇒ P2 etc.)

Achieving Privacy as well as Utility

- Utility is about "finding correlations in data"
- Privacy is about "removing correlations in data"
- Ways to remove "correlations"
  - Anonymise data (Alice $\Rightarrow$ P1, Bob $\Rightarrow$ P2 etc.)
  - Add "noise" (add spurious rows to column)

Achieving Privacy as well as Utility

- Utility is about "finding correlations in data"
- Privacy is about "removing correlations in data"
- Ways to remove "correlations"
  - Anonymise data (Alice $\Rightarrow$ P1, Bob $\Rightarrow$ P2 etc.)
  - Add "noise" (add spurious rows to column)
  - Remove "sensitive" columns

Achieving Privacy as well as Utility

- Utility is about "finding correlations in data"
- Privacy is about "removing correlations in data"
- Ways to remove "correlations"
  - Anonymise data (Alice $\Rightarrow$ P1, Bob $\Rightarrow$ P2 etc.)
  - Add "noise" (add spurious rows to column)
  - **Remove "sensitive" columns** (✓)

Name	Roll Number	Department	Program	Income Range
Bob	1003	ME	BT	50K - 100K
Alice	1002	CSE	MS	>500K
John	1004	PHY	MT	100K - 350K
Mary	1005	CSE	PHD	50K - 100K
José	1006	MTH	BS	350 - 500K

Department	Program	Income Range
ME	BT	50K - 100K
CSE	MS	>500K
PHY	MT	100K - 350K
CSE	PHD	50K - 100K
MTH	BS	350 - 500K

Name	Roll Number	Department	Program	Income Range
Bob	1003	ME	BT	50K - 100K
Alice	1002	CSE	MS	>500K
John	1004	PHY	MT	100K - 350K
Mary	1005	CSE	PHD	50K - 100K
José	1006	MTH		

The correlation between individuals and their incomes has been removed

Department	Program	Income Range
ME	BT	50K - 100K
CSE	MS	>500K
PHY	MT	100K - 350K
CSE	PHD	50K - 100K
MTH	BS	350 - 500K

Name	Roll Number	Department	Program	Income Range
Bob	1003	ME	BT	50K - 100K
Alice	1002	CSE	MS	>500K
John	1004	PHY	MT	100K - 350K
Mary	1005	CSE	PHD	50K - 100K
		MTH	BS	350 - 500K

But some utility of the data is also "lost" (e.g. selecting financially weaker students for "scholarships")

Department	Program	Income Range
ME	BT	50K - 100K
CSE	MS	>500K
PHY	MT	100K - 350K
CSE	PHD	50K - 100K
MTH	BS	350 - 500K

Achieving Privacy as well as Utility

- Utility is about "finding correlations in data"
- Privacy is about "removing correlations in data"
- Ways to remove "correlations"
  - Anonymise data (Alice ⇒ P1, Bob ⇒ P2 etc.)
  - Add "noise" (add spurious rows to column)
  - Remove "sensitive" columns
- Irrespective of what options we choose, the data almost always uses "some utility"

Achieving Privacy as well as Utility

- Utility is about "finding correlations in data"
- Privacy is about "removing correlations in data"
- Ways to remove "correlations"
  - Anonymise data (Alice ⇒ P1, Bob ⇒ P2 etc.)
  - Add "noise" (add spurious rows to column)
  - Remove "sensitive" columns
- Irrespective of what options we choose, the data almost always uses "some utility"
- So, there is a *trade-off* here, and we need to find a mid-way out of it !

The Trade-off Model

Understanding a simple solution to the problem

Pruning the data to achieve Privacy

- Let us assume we have a table with *n* attributes and *m* rows

Pruning the data to achieve Privacy

- Let us assume we have a table with n attributes and m rows
- Also, there are some set of attributes which, if present together in a table, can result in a potential breach of privacy
    - Last example – (Name, Income Range), (Roll Number, Income Range) etc.

Pruning the data to achieve Privacy

- Let us assume we have a table with *n* attributes and *m* rows
- Also, there are some set of attributes which, if present together in a table, can result in a potential breach of privacy
  - Last example – (Name, Income Range), (Roll Number, Income Range) etc.
- If we divide this table into multiple *partitions*, with each partition containing some attributes of the table, we can essentially remove some instances of possible privacy breach
- We cater to a class of applications, which use data for *classification* purposes – so the class attribute (not counted in *n*) is copied to all partitions, to make sure that the partition is useful for classification

age	**workclass**	**marital-status**	**race**	***class***
39	State-gov	Never-married	White	*<=50K*
49	Self-emp-inc	Married-civ-spouse	White	*>50K*
28	Private	Married-civ-spouse	Other	*<=50K*
35	Private	Divorced	White	*>50K*
38	Private	Divorced	White	*<=50K*
53	Local-gov	Never-married	White	*<=50K*
28	Private	Married-civ-spouse	Black	*<=50K*
37	Private	Married-civ-spouse	Black	*>50K*
37	Private	Married-civ-spouse	White	*<=50K*
49	Private	Married-spouse-absent	Black	*<=50K*
38	Federal-gov	Married-civ-spouse	White	*>50K*
42	Private	Married-civ-spouse	White	*>50K*

Table 1. An excerpt from the UCI Adult dataset

age	**marital-status**	**race**	***class***
35	Divorced	White	*>50K*
38	Divorced	White	*<=50K*
53	Never-married	White	*<=50K*
49	Married-civ-spouse	Black	*<=50K*
42	Married-civ-spouse	White	*>50K*

age	**workclass**	***class***
53	Local-gov	*<=50K*
28	Private	*<=50K*
35	Private	*>50K*
37	Private	*<=50K*
39	State-gov	*<=50K*
49	Private	*<=50K*

race	***class***
White	*<=50K*
Black	*<=50K*
White	*>50K*
Other	*<=50K*

age	***class***
37	*>50K*
49	*>50K*
38	*<=50K*
42	*>50K*
38	*>50K*

Figure 1. Some partitions of the dataset in Table 1

Picking a partition to use

- Let us assume that we would like to use a partition of the original data for the classification task, instead of the whole data

Picking a partition to use

- Let us assume that we would like to use a partition of the original data for the classification task, instead of the whole data
- The question is – Which partition to use? More specifically,
  - What sized partition is "good enough"? Since *Partition Size* $\in [1, n]$
  - Among partitions of the same size, how choosing one is different from other?

Picking a partition to use

- Let us assume that we would like to use a partition of the original data for the classification task, instead of the whole data
- The question is – Which partition to use? More specifically,
  - What sized partition is "good enough"? Since *Partition Size* $\in [1, n]$
  - Among partitions of the same size, how choosing one is different from other?
- We can use statistical analysis with sophisticated metrics to analyse privacy and utility of each partition, and pick a partition

Picking a partition to use

- Let us assume that we would like to use a partition of the original data for the classification task, instead of the whole data
- The question is – Which partition to use? More specifically,
  - What sized partition is "good enough"? Since *Partition Size* $\in [1, n]$
  - Among partitions of the same size, how choosing one is different from other?
- We can use statistical analysis with sophisticated metrics to analyse privacy and utility of each partition, and pick a partition
- Or, we can attempt an engineering solution via an experimental setup, that doesn't require in-depth statistical knowledge

Picking a partition to use

- Let us assume that we would like to use a partition of the original data for the classification task, instead of the whole data
- The question is – Which partition to use? More specifically,
  - What sized partition is "good enough"? Since *Partition Size* $\in [1, n]$
  - Among partitions of the same size, how choosing one is different from other?
- We can use statistical analysis with sophisticated metrics to analyse privacy and utility of each partition, and pick a partition
- **Or, we can attempt an engineering solution via an experimental setup, that doesn't require in-depth statistical knowledge (✓)**

Trade-off Model

- *Input*
  - A Table *T*, with *n* attributes and *m* rows; additionally, the table has another attribute called the *class* attribute (making total columns *n+1*)
  - Partition Size, *p* : An integer between *1* and *n*
  - Classification Objective, *O* : The technique to be used for classification of data
  - Privacy Exceptions, *PE* : A possibly empty list of attribute combinations, which may pose a risk to privacy; the size of a combination can be at max *p*
  - Utility Exceptions, *UE* : A possible empty list of attribute combinations, which are desirable in the output partitions; the size of a combination can be at max *p*
  - Optional metric *M* to sort the results (e.g. Accuracy, False Positive Rate etc.)
- *Output*
  - A list of partitions, *P*, sorted by *M*; each partition contains *p* attributes (+ *class*)
  - A list of values for *M*, corresponding to each partition in *P*

Input to the model

partition size = 2;
privacy exceptions = { (age, workclass) };
learning objective = Classification(NaiveBayes);

Output from the model

{age, race}	**58.333333333333336%** (✓)
{age, marital-status}	33.333333333333336%
{workclass, marital-status}	33.333333333333336%
{marital-status, race}	33.333333333333336%
{workclass, race}	25.0%

Overall methodology

- Step 1: Create a list of partitions, possible for a given partition size, that do not contain any combinations supplied in *PE*
  - For example, for *p* = 2 : [{*age*, *marital-status*}, {*age*, *race*}, {*workclass*, *marital-status*}, {*workclass*, *race*}, {*marital-status*, *race*}]
- Step 2: Invoke a task, applying *O* over all selected partitions, and note down the value of *M* produced by each task
  - For example, for *Naïve Bayes Classification* and Metric Classification Accuracy, compute and store entries like [{*age*, *marital-status*} ⇒ *33.333333%*]
- Step 3: Sort the list of partitions, by their corresponding *M* values, to produce *P*

Engineering additions

Building a *practical* prototype for the model

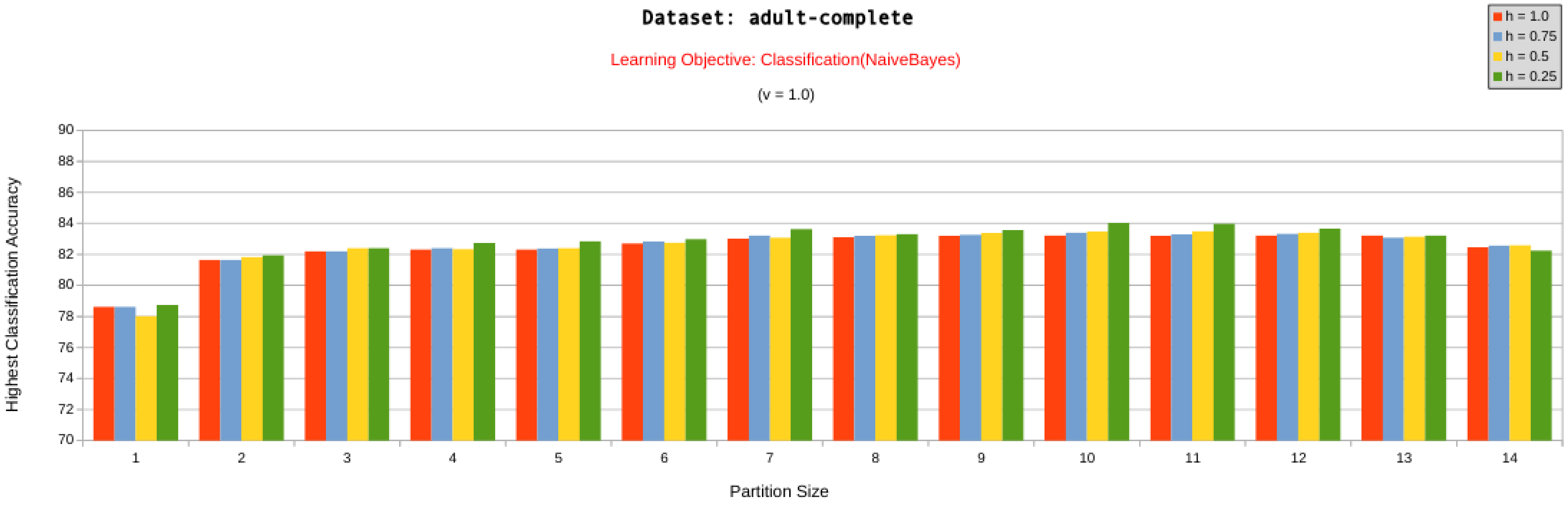
Reducing the number of possible partitions

- The function that actually determines the number of partitions is the *Combinations function*, $C(n, p)$
  - For n = 25, p = 10, the number of possible partitions is **3,268,760** !!!
- Clearly, we cannot run the classification tasks for all these partitions in a practical solution
- So, we added another "engineering" parameter to the model – called the *Vertical Expense*, $v \in (0, 1]$
- It defines the proportion of possible partitions, that should be tried out for experiments
  - For example (v = 0.5) ⇒ "*try only 50% of possible partitions*"

Fastening the individual classification tasks

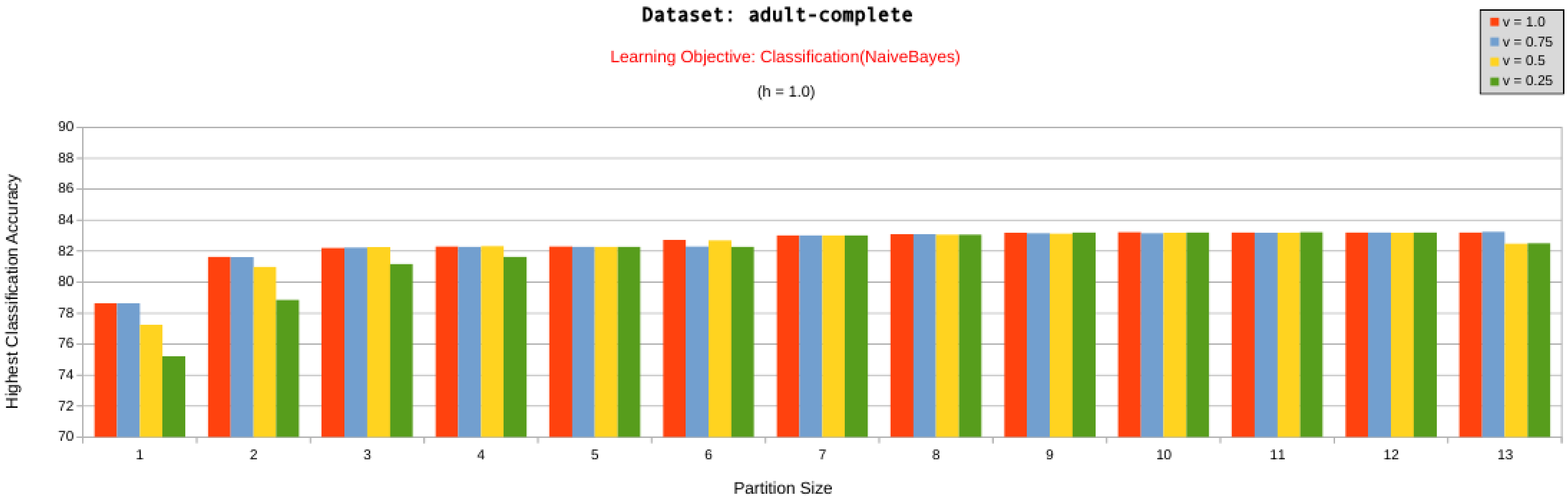
- The experiments we perform are *indicative* - i.e. they are best-effort approximations to a larger, complex problem
- If the original dataset contains a lot of rows (say a million !!), running so many classification tasks will be extremely time consuming
- Similar to v, that can reduce the number of partitions that will be tried out, we define another engineering parameter, called the Horizontal Expense $h \in (0, 1]$
- It defines the proportion of rows from the original dataset to be used in individual classification tasks
  - For example ($h = 0.1$) ⇒ "*use any 10% of the rows for individual tasks*"

Effects of changing Horizontal Expense



(a) Varying horizontal expense, keeping vertical expense constant

Effects of changing Vertical Expense



(b) Varying vertical expense, keeping horizontal expense constant

Thanks for your time !!

Questions?