Privacy-Aware Machine Learning Systems

Borja Balle

research cambridge

Data is the New Oil

The Economist, May 2017

research cambridge

The Importance of (Data) Privacy

Universal declaration of human rights

Article 12. No one shall be subjected to arbitrary interference with his **privacy**, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.

#DeleteFacebook

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

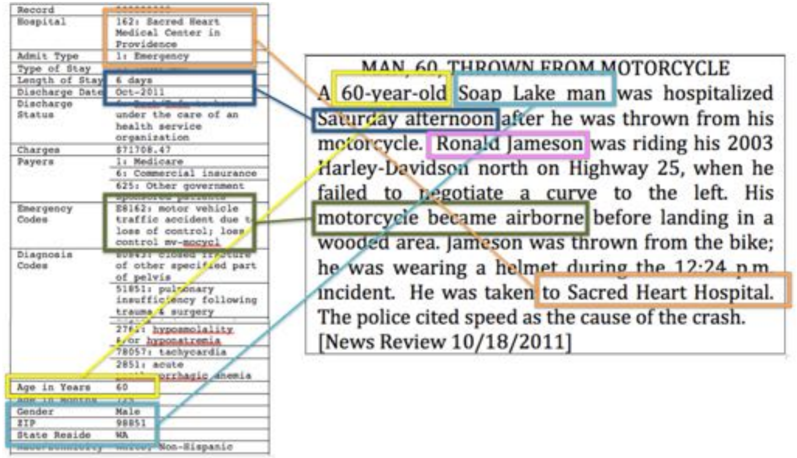
of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

Anonymization Fiascos

Vijay Pandurangan. *tech.vijayp.ca*, 2014

"Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset)"
A. Narayanan & V. Shmatikov. *Security and Privacy, 2008*



"Only You, Your Doctor, and Many Others May Know"
L. Sweeney. *Technology Science, 2015*

research cambridge

Privacy Risks in Machine Learning

Membership Inference Attacks Against Machine Learning Models

Reza Shokri
Cornell Tech

Marco Stronati*
INRIA

Congzheng Song
Cornell

Vitaly Shmatikov
Cornell Tech

***Abstract*—We quantitatively investigate how machine learning models leak information about the individual data records on which they were trained. We focus on the basic membership inference attack: given a data record and black-box access to a model, determine if the record was in the model's training dataset. To perform membership inference against a target model, we make adversarial use of machine learning and train our own inference model to recognize differences in the target model's predictions on the inputs that it trained on versus the inputs that it did not train on.**

We empirically evaluate our inference techniques on classification models trained by commercial "machine learning as a service" providers such as Google and Amazon. Using realistic datasets and classification tasks, including a hospital discharge dataset whose membership is sensitive from the privacy perspective, we show that these models can be vulnerable to membership inference attacks. We then investigate the factors that influence this leakage and evaluate mitigation strategies.

Security and Privacy, 2017

The Secret Sharer:
Measuring Unintended Neural Network Memorization & Extracting Secrets

Nicholas Carlini
University of California, Berkeley

Chang Liu
University of California, Berkeley

Jernej Kos
National University of Singapore

Úlfar Erlingsson
Google Brain

Dawn Song
University of California, Berkeley

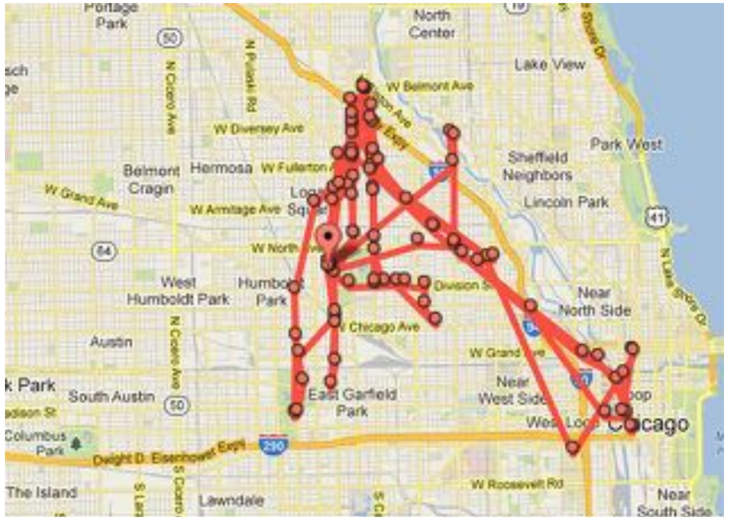
This paper presents *exposure*, a simple-to-compute metric that can be applied to any deep learning model for measuring the memorization of secrets. Using this metric, we show how to extract those secrets efficiently using black-box API access. Further, we show that unintended memorization occurs early, is not due to over-fitting, and is a persistent issue across different types of models, hyperparameters, and training strategies. We experiment with both real-world models (e.g., a state-of-the-art translation model) and datasets (e.g., the Enron email dataset, which contains users' credit card numbers) to demonstrate both the utility of measuring exposure and the ability to extract secrets.

Finally, we consider many defenses, finding some ineffective (like regularization), and others to lack guarantees. However, by instantiating our own differentially-private recurrent model, we validate that by appropriately investing in the use of state-of-the-art techniques, the problem can be resolved, with high utility.

ArXiv, 2018

What Makes Privacy Difficult?

High-dimensional data



Side information

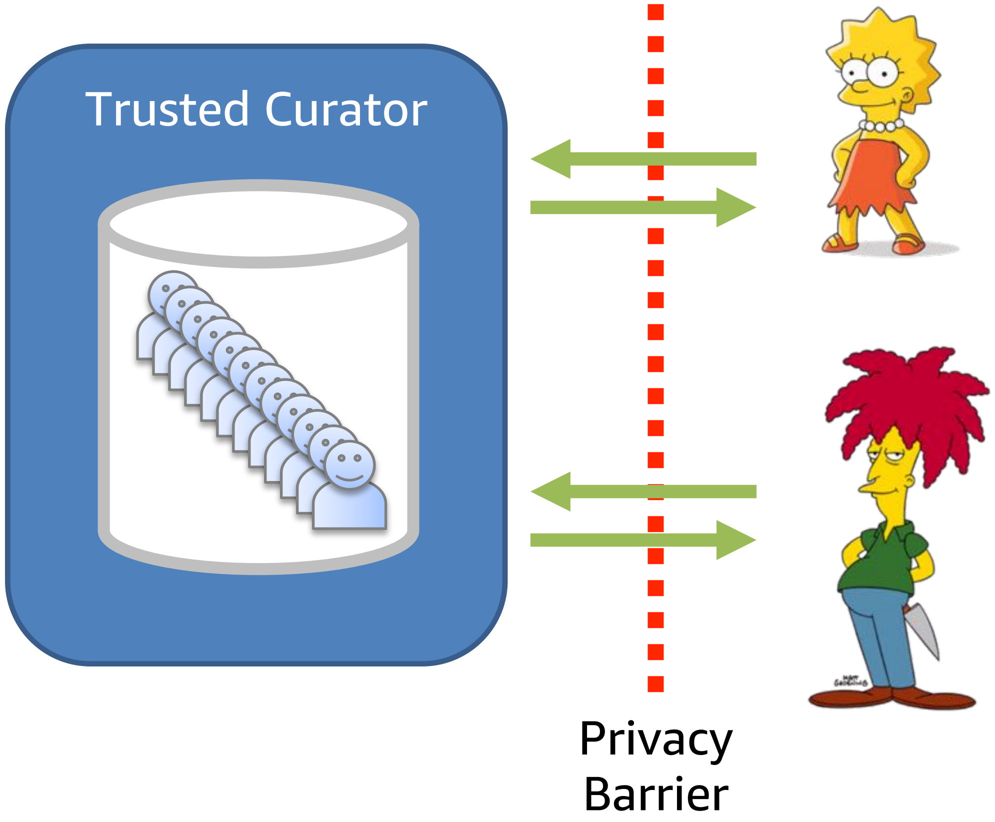
Privacy Enhancing Technologies (PETS)

- Initially a sub-field of applied cryptography
  - Now percolating into databases, machine learning, statistics, etc.

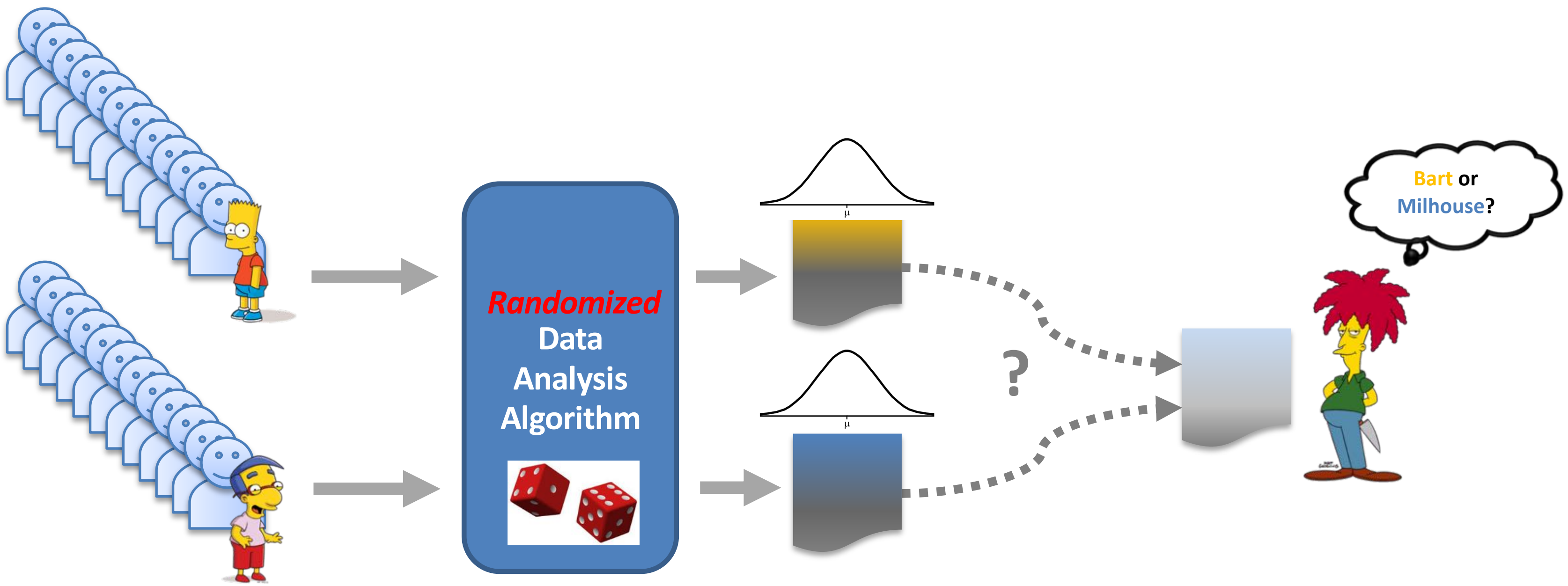
- Privacy-preserving **release** (eg. differential privacy)
  - Release statistics/models/datasets while preventing reverse-engineering of the original data

- Privacy-preserving **computation** (eg. secure multi-party computation)
  - Perform computations on multi-party data without *ever* exchanging the inputs in plaintext

Privacy-Preserving Release



Differential Privacy: Informal Definition

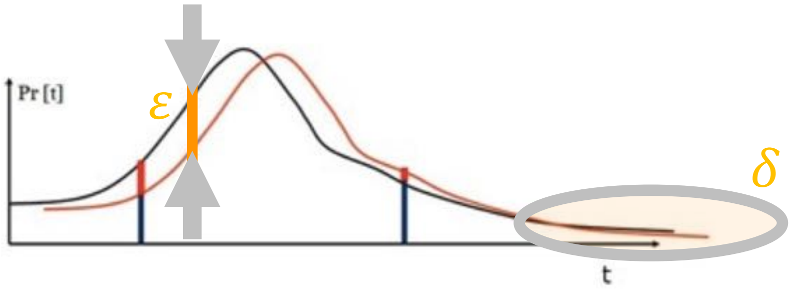


Differential Privacy

[DMNS'06; Godel Prize 2017]

A randomized algorithm $M : X^n \rightarrow Y$ satisfies differential privacy with parameter ε if for any pair of datasets x and x' differing in a single row and for any possible output y, the following inequality is satisfied:

$$\mathbb{P}[M(x) = y] \leq e^{\varepsilon}\mathbb{P}[M(x') = y]$$



... approximate differential privacy with parameters (ε, δ) ... set of outputs E ...

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

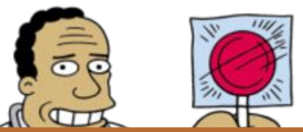
Fundamental Properties of Differential Privacy

- Compositionality
  - Enables rigorous engineering through modularity
- Quantifiable
  - Amenable to mathematical analysis, continuous instead of black-or-white
- Robust to side knowledge
  - Protects even in the event of collusions and side information

Multi-Party Data Analysis

Treatment Outcome	Medical Data		
	Attr. 1	Attr. 2	…
-1.0	0	54.3	…
1.5	1	0.6	…
-0.3	1	16.0	…
0.7	0	35.0	…
3.1	1	20.2	…

The Trusted Party "Solution"



The **Trusted Party** assumption:

- Introduces a **single point of failure** (with disastrous consequences)
- Relies on **weak incentives** (especially when private data is valuable)
- Requires **agreement** between all data providers

=> Useful but unrealistic. Maybe **can be simulated**?

Party

Receives plain-text data, runs algorithm, returns result to parties

Secure Multi-Party Computation (MPC)

Public: $f(x_1, x_2, \ldots, x_p) = y$

Private: (party i) x_i

Goal: Compute f in a way that each party learns y (and nothing else!)

Tools: Oblivious Transfers (OT), Garbled Circuits (GC), Homomorphic Encryption (HE), etc

Guarantees: Honest but curious adversaries, malicious adversaries, computationally bounded adversaries, collusions

Challenges and Trade-offs

- Protocols: out of the box vs. tailored
- Threat models: semi-honest vs. malicious
- Interaction: off-line vs. on-line
- Trusted external parties: speed vs. privacy
- Scalability: amount of data, dimensions, # parties

In This Talk…

Part I: Privacy-Preserving Distributed Linear Regression on High-Dimensional Data

PETS 2017, with ***Adria Gascon, Phillipp Schoppmann, Mariana Raykova, Jack Doerner, Samee Zahur, and David Evans***

Part II: Private Nearest Neighbors Classification in Federated Databases

Preprint, with ***Adria Gascon and Phillipp Schoppmann***

Linear Regression - Overview

Features:

- Vertically partitioned data
- Scalable to millions of records and hundreds of dimensions
- Open source implementation
  `https://github.com/schoppmp/linreg-mpc`

Tools:

- Several standard MPC constructions (GC, OT, SS, …)
- Efficient private inner product protocols
- Conjugate gradient descent robust to fixed-point encodings

Functionality: Multi-Party Linear Regression

Training Data

$$X = [X_1 \ X_2] \in \mathbb{R}^{n \times d}$$

$$Y \in \mathbb{R}^n$$

Private Inputs

Party 1: X_1, Y

Party 2: X_2

Linear Regression

$$\min_{\theta \in \mathbb{R}^d} \|Y - X\theta\|^2 + \lambda\|\theta\|^2$$

(optimization)

$$(X^\top X + \lambda I)\theta = X^\top Y$$

(closed-form solution)

Aggregation and Solving Phases

Aggregation

$$A = X^\top X + \lambda I$$

$$b = X^\top Y$$

$$\mathcal{O}(nd^2)$$

$$X^\top X = \begin{bmatrix} X_1^\top X_1 & \boxed{X_1^\top X_2} \\ \boxed{X_2^\top X_1} & X_2^\top X_2 \end{bmatrix}$$

(cross-party products)

Solving

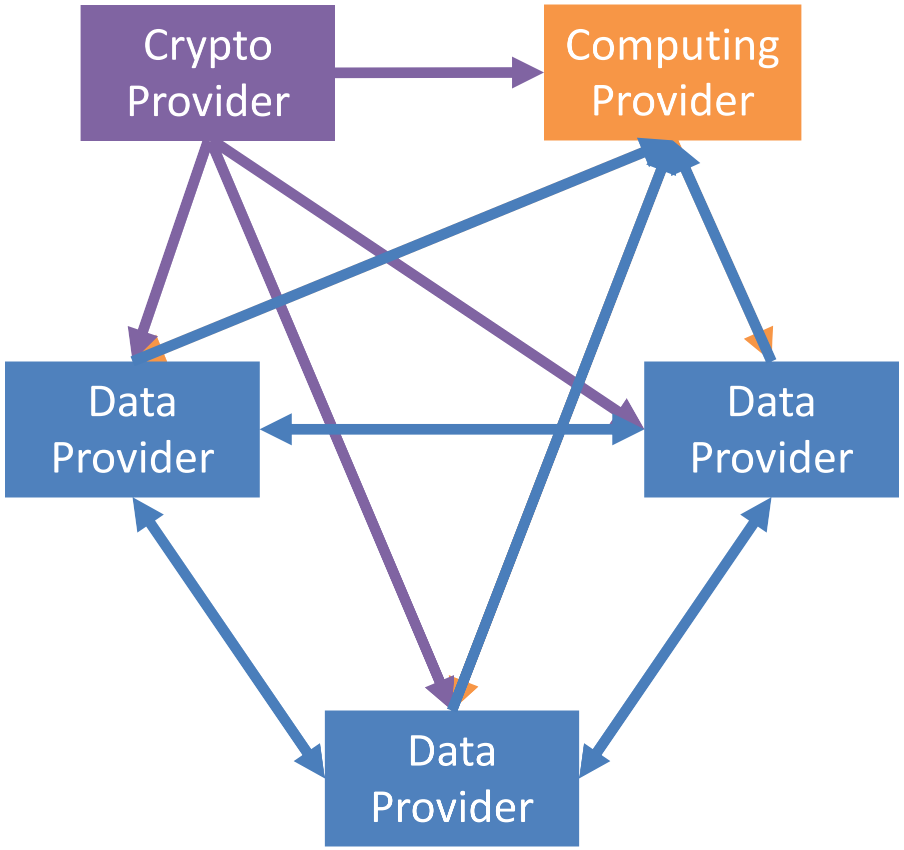
$$\theta = A^{-1} b$$

$\mathcal{O}(d^3)$ (eg. Cholesky)

Approximate iterative solver $\mathcal{O}(kd^2)$

(eg. k-CGD)

Protocol Overview



Alternative: CrP and CoP simulated by non-colluding parties

Aggregation Phase

1. CrP distributes correlated randomness
2. DPs run multiple inner product protocols to get additive share of (A,b)

Solving Phase

3. CoP get GC for solving linear system from CrP
4. DPs send garbled shares of (A,b) to CoP
5. CoP executes GC and returns solution to DPs

Aggregation Phase – Arithmetic Secret Sharing

$$X_1^\top X_2 \longrightarrow f(x_1, x_2) = \langle x_1, x_2 \rangle$$

(matrix product) (inner product b/w columns)

Party 1

x_1

a, c

$x_2 - b$

$s_1 = x_1 \cdot (x_2 - b) - c$

$a \cdot b = c + d$

oblivious transfer / 3rd party

Party 2

x_2

b, d

$x_1 + a$

$s_2 = (x_1 + a) \cdot b - d$

$$s_1 + s_2 = x_1 \cdot x_2$$

Solving Phase – Garbled Circuits

$$(A_i, b_i)$$

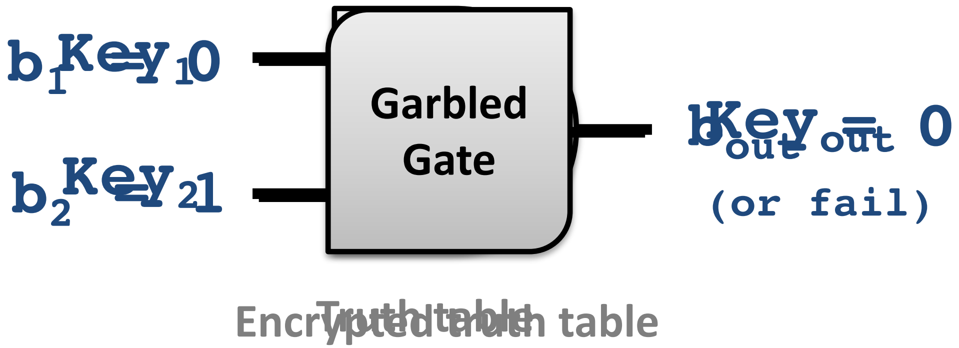
(party i's input: arithmetic share)

$$A = \sum_i A_i \quad b = \sum_i b_i$$

$$A\theta = b$$

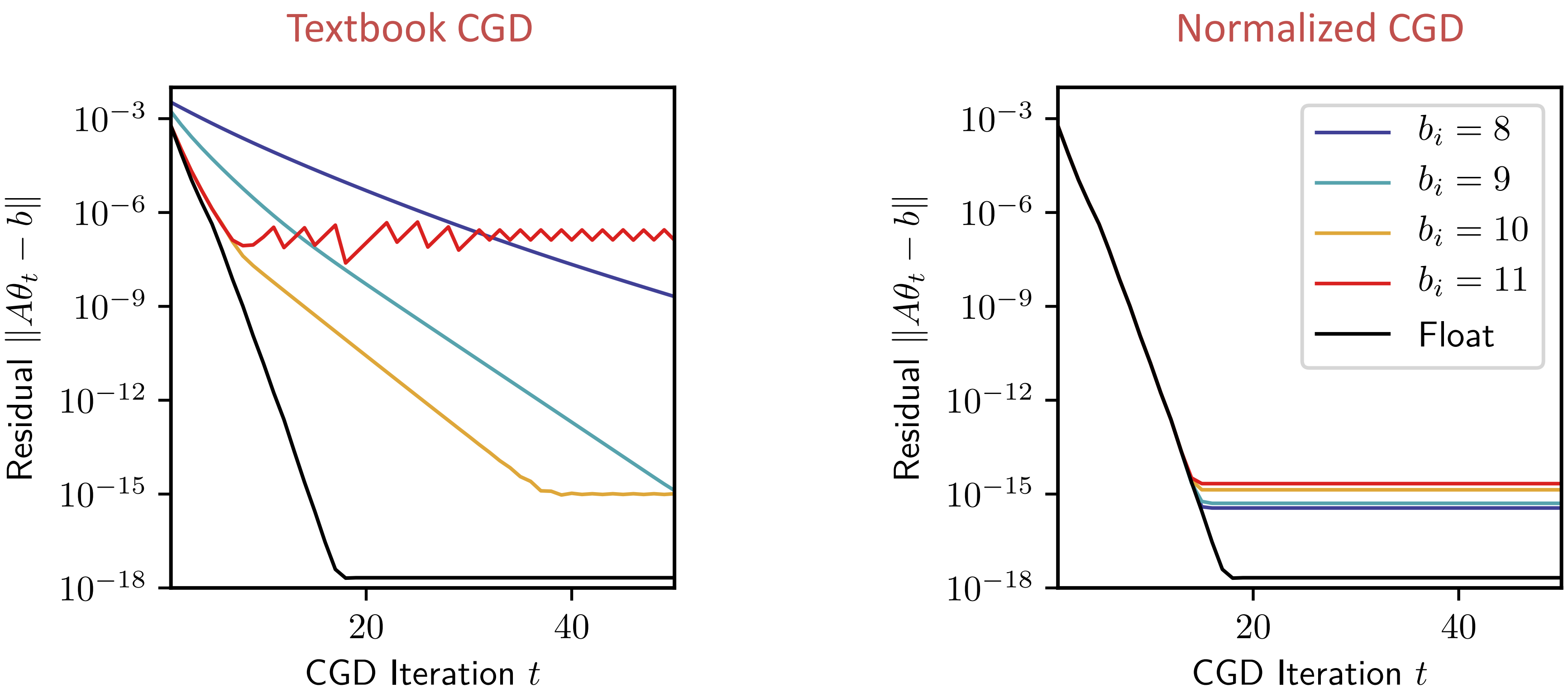
(PSD linear system)

Solved with
Conjugate Gradient Descent (CGD)



Year	Device / Paper	32 bit floating point multiplication (ms)
1961	IBM 1620E	**17.7**
1980	Intel 8086 CPU (software)	**1.6**
1980	Intel 8087 FPU	**0.019**
2015	Pullonen et al. @ FC&DS	**38.2**
2015	Demmler et al. @ CCS	**9.2**

Fixed-point + Conjugate Gradient Descent



Total number of bits = $b_i + b_f + 1$

b_i = number of integer bits

b_f = number of fractional bits

Experimental Results

Aggregation Phase

n	d	Number of parties: 2		3		5	
		OT	TI	OT	TI	OT	TI
$5 \cdot 10^4$	20	1m50s	1s	1m32s	2s	1m7s	2s
$5 \cdot 10^4$	100	42m12s	25s	34m39s	32s	24m58s	37s
$5 \cdot 10^5$	20	18m18s	15s	14m29s	18s	12m10s	21s
$5 \cdot 10^5$	100	7h3m56s	4m47s	5h20m52s	6m1s	4h17m8s	6m58s
$1 \cdot 10^6$	100	-	10m1s	-	12m42s	-	14m48s
$1 \cdot 10^6$	200	-	39m16s	-	49m56s	-	59m22s

Solving Phase

Name	d	n	Optimal RMSE	FP-CGD (32 bits) time	FP-CGD (32 bits) RMSE	Cholesky (32 bits) time	Cholesky (32 bits) RMSE
Student Performance	30	395	4.65	19s	4.65 (-0.0%)	5s	4.65 (-0.0%)
Auto MPG	7	398	3.45	2s	3.45 (-0.0%)	0s	3.45 (-0.0%)
Communities and Crime	122	1994	0.14	4m27s	0.14 (0.3%)	4m35s	0.14 (-0.0%)
Wine Quality	11	4898	0.76	3s	0.76 (-0.0%)	0s	0.80 (4.2%)
Bike Sharing Dataset	12	17379	145.06	4s	145.07 (0.0%)	1s	145.07 (0.0%)
Blog Feedback	280	52397	31.89	24m5s	31.90 (0.0%)	53m24s	32.19 (0.9%)
CT slices	384	53500	8.31	44m46s	8.34 (0.4%)	2h13m31s	8.87 (6.7%)
Year Prediction MSD	90	515345	9.56	4m16s	9.56 (0.0%)	3m50s	9.56 (0.0%)
Gas sensor array	16	4208261	90.33	48s	95.05 (5.2%)	42s	95.06 (5.2%)

Related Work

Ref	Crypto	Solver	n (max)	d (max)	Iterative	Bottleneck
[1]	HE	Newton	50K	22	Local (40)	Computation
[2]	HE+GC	Cholesky	10M	14	No	Both
[3]	SS	CGD	10K	10	Network (10)	Network
*	SS+GC	CGD	1M	500	Local (20)	Computation
[4]	HE	GD-VWT	97	8	Local (4)	Computation
[5]	SS	SGD	1M	784	Network (100-1000)	Network

[1] Hall et al. (2011). Secure multiple linear regression based on homomorphic encryption. Journal of Official Statistics.

[2] Nikolaenko et al. (2013). Privacy-preserving ridge regression on hundreds of millions of records. In Security and Privacy (SP).

[3] Bogdanov et al. (2016). Rmind: a tool for cryptographically secure statistical analysis. IEEE Transactions on Dependable and Secure Computing.

[4] Esperanca et al. (2017). Encrypted Accelerated Least Squares Regression. In AISTATS.

[5] Mohassel et al. (2017). SecureML: A System for Scalable Privacy-Preserving Machine Learning. In Security and Privacy (SP).

Linear Regression - Conclusion

Summary

- Full system is accurate and fast, available as open source
- Scalability requires hybrid MPC protocols and non-trivial engineering
- Robust fixed-point CGD inside GC has many other applications

Extensions

- Security against malicious adversaries
- Classification with quadratic loss
- Kernel ridge regression
- Differential privacy on the covariance / at the output

Future Work

- Models without a closed-form solution (eg. logistic regression, DNN)
- Library of re-usable ML components, complete data science pipeline

In This Talk…

Part I: Privacy-Preserving Distributed Linear Regression on High-Dimensional Data

PETS 2017, with ***Adria Gascon, Phillipp Schoppmann, Mariana Raykova, Jack Doerner, Samee Zahur, and David Evans***

Part II: Private Nearest Neighbors Classification in Federated Databases

Preprint, with ***Adria Gascon and Phillipp Schoppmann***

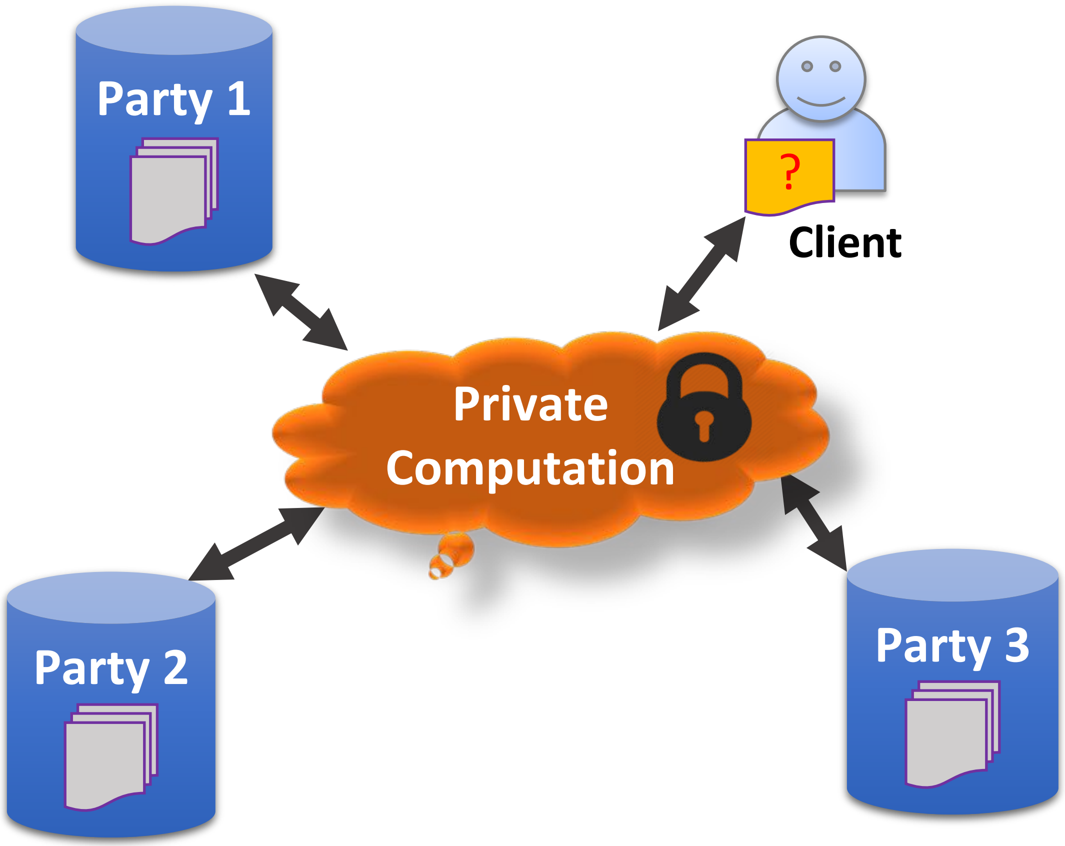
Document Classification - Overview

Setup:

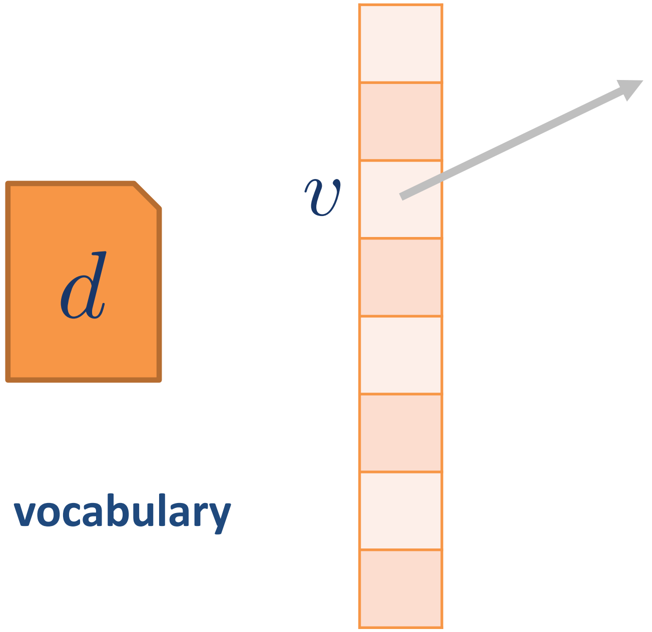
- Federated database held by multiple (untrusting) parties
- Database and client's document should be kept private
- k-NN classification with TF-IDF features and cosine similarity

Contributions:

- Multi-party computational DP protocol
  - DP computation of IDFs
  - MPC protocol for sparse inner products
- Privacy against arbitrary collusions



Document Classification with Nearest Neighbors



$$\psi_d(v) = \mathrm{tf}_d(v) \cdot \mathrm{idf}_Z(v)$$

$$\mathrm{idf}_Z(v) \approx \log \frac{|Z|}{|Z_v|}$$

Z **document dataset**

V **vocabulary**

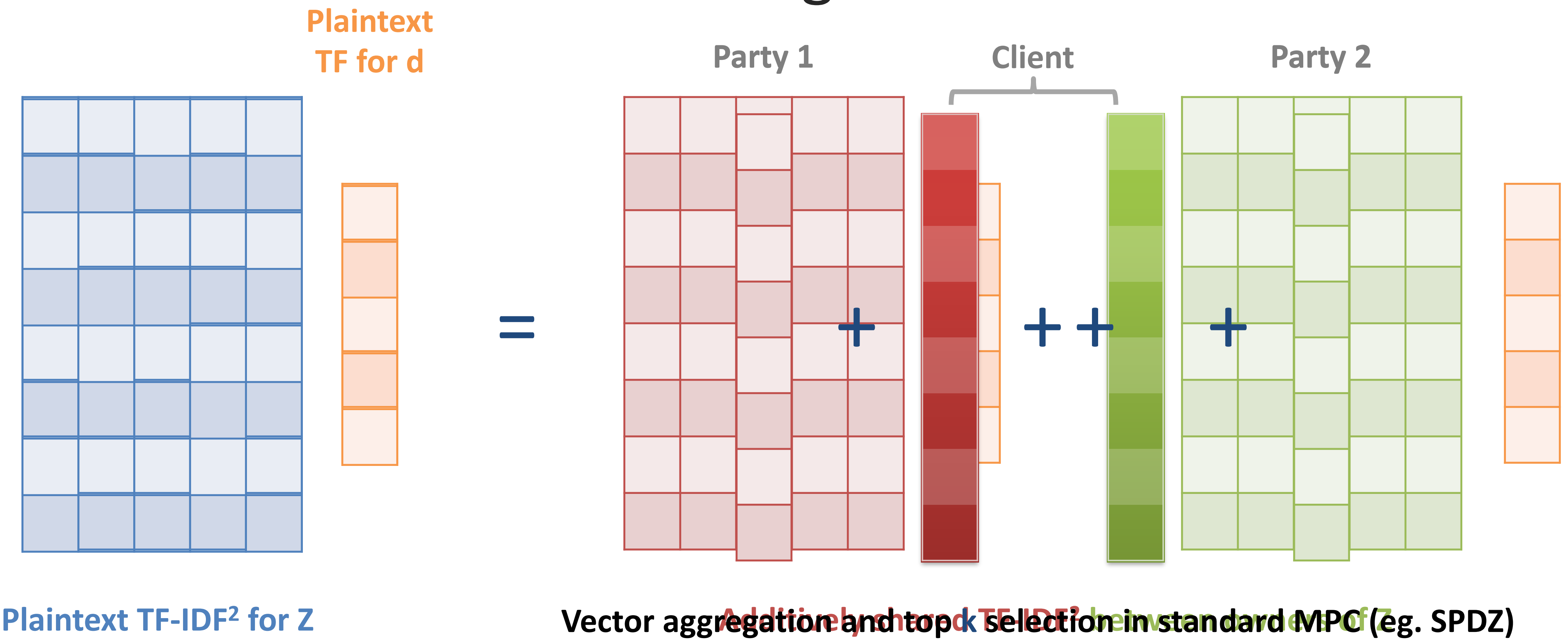
$$\psi_d \in \mathbb{R}^{|V|}$$

1. For each x in Z compute the score

$$\mathrm{score}(d, x) = \frac{\langle \psi_d, \psi_x \rangle}{\|\psi_d\| \|\psi_x\|}$$

2. Label d by majority on top k scores

Secret Sharing Baseline



Vector aggregation and top k selection in standard MPC (eg. SPDZ)

Pros: Shares can pre-computed, inner product protocol

Cons: Additive shares destroy sparsity

Sparse Protocol

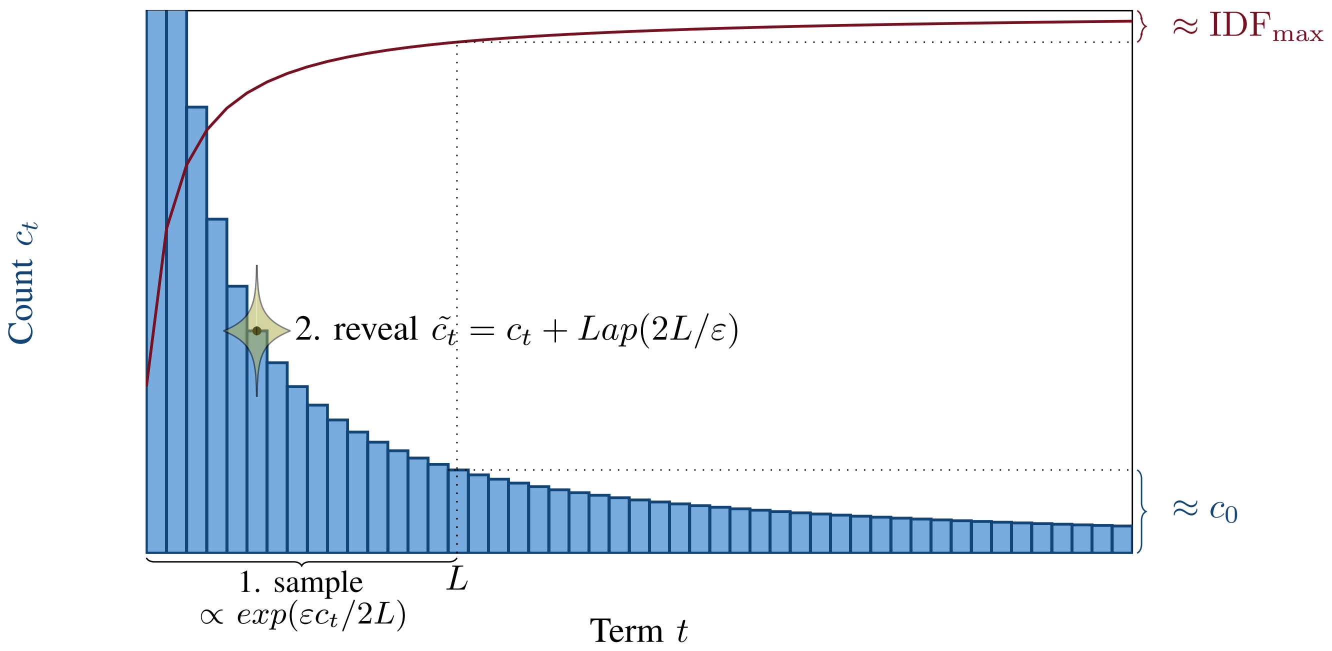
1. Compute IDFs on dataset Z using differential privacy
   - Implement Laplace and Exponential mechanism inside MPC protocol (eg. SPDZ). Yields *Computational Differential Privacy* guarantees.
2. Use custom sparse matrix-vector multiplication protocol
   - Run between client and each data provider
   - Produce arithmetic shares as output
3. Aggregate shares to get scores and select top k
   - Same as in baseline protocol

Computing IDFs with Differential Privacy

Algorithm 1: DP IDFs

Input: Public: n, $\mathcal{V}$, c_0, L, ε_0
Input: Private: Counts $\{|Z_i|_v\}_{v\in\mathcal{V}}$ for $i \in [n]$
Output: Privatized values $\{\tilde{c}_v\}_{v\in\mathcal{V}}$

foreach $v \in \mathcal{V}$ **do**
  Compute $c_v = \sum_{i=1}^{n} |Z_i|_v$
end
for $\ell = 1, \ldots, L$ **do**
  Sample $v \in \mathcal{V}$ with probability $\propto \exp(\varepsilon_0 c_v)$
  Sample η from $\mathsf{Lap}(1/\varepsilon_0)$
  Release $\tilde{c}_v = c_v + \eta$
  Remove v from $\mathcal{V}$
end
For each $v \in \mathcal{V}$ release $\tilde{c}_v = c_0$



Theorem 2. *For any $\varepsilon_0 \in (0, 0.9]$ and $\delta \in [0,1]$ the Algorithm 1 is (ε, δ)-DP with*

$$\varepsilon = \min\left\{2L\varepsilon_0, 2L\varepsilon_0^2 + \sqrt{4L\varepsilon_0^2 \log(1/\delta)}\right\} .$$

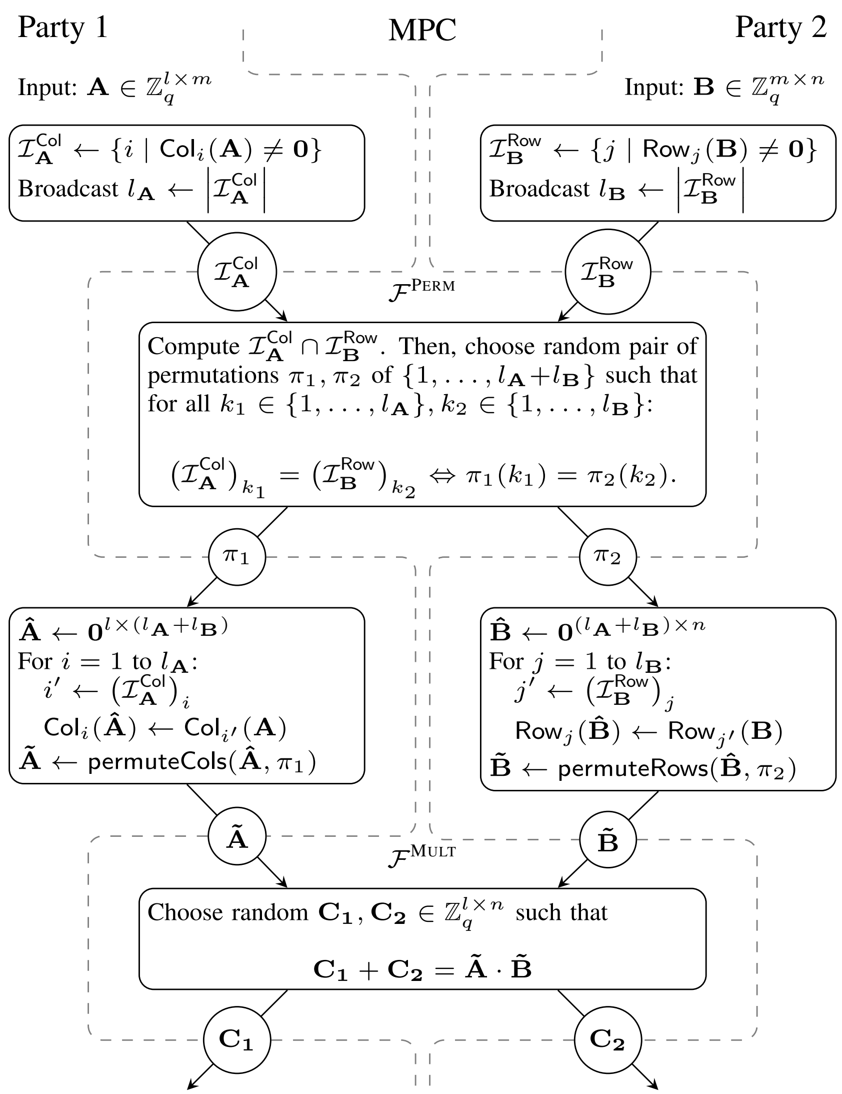
Theorem 3. *Let $c_0 = \Theta(\sqrt{m})$. If m is large enough, then with high probability we have*

$$\frac{\|\phi_{\mathrm{idf}} - \tilde{\phi}_{\mathrm{idf}}\|_1}{\|\phi_{\mathrm{idf}}\|_1} \le \tilde{O}\left(\frac{L}{V}\frac{1}{\varepsilon_0 m} + \left(1 - \frac{L}{V}\right)\log(m)\right) .$$

research cambridge

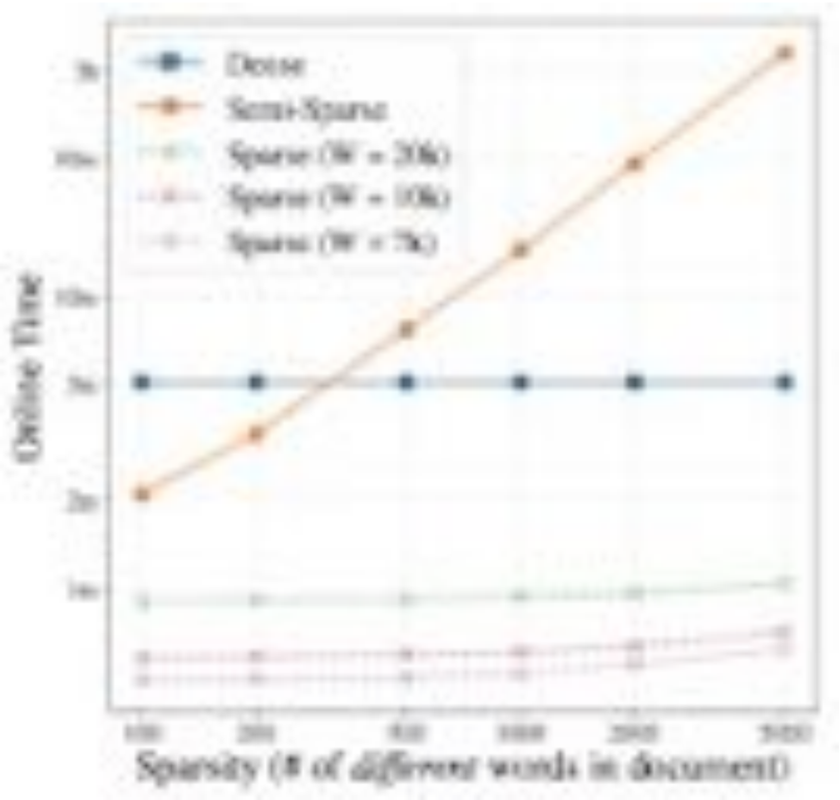
Private Sparse Multiplication

- Idea: Reduce sparse multiplication to non-sparse multiplication
- How: Find common non-zero coefficients and restrict to these coordinates
- In MPC: Private set intersection
- Leakage: Upper bound on number of non-zeros

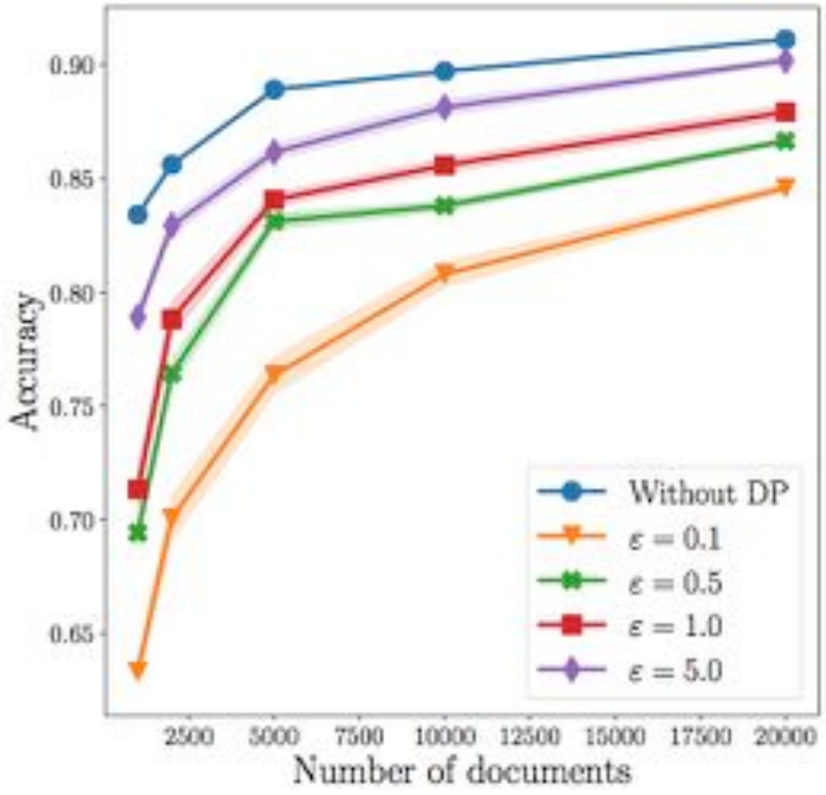


Illustrative Experiments

Speed (vs. sparsity)



Accuracy (vs. privacy)



Document Classification - Conclusion

Conclusions

- Non-parametric models are challenging from the privacy point of view
- Changes in privacy assumptions enable different solutions
- Protocols with different speed/privacy/accuracy trade-offs
- Sparse matrix-vector multiplication is an important primitive for PMPML

Future Work

- Better DP algorithms for feature extraction
- Other features instead of TF-IDF
- Full open source implementation

Take Home Points

- Re-visiting basic ML algorithms from an MPC+DP perspective yields important insights for tackling more complex problems
- ML can motivate the development of new MPC primitives (eg. linear algebra)
- Rich toolbox, plenty of unexplored combinations
- Trade-offs: privacy/speed/accuracy
- Genuine interdisciplinary effort