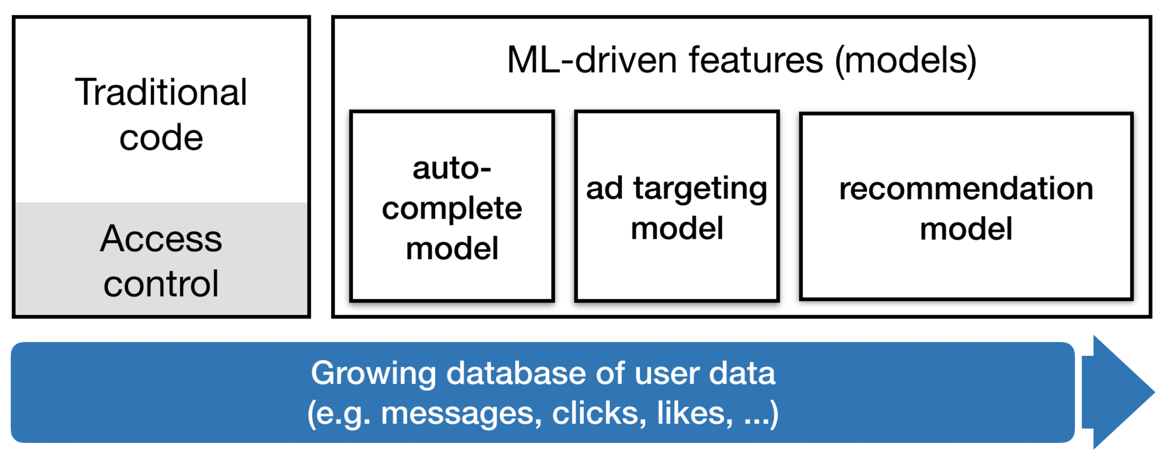
Privacy Accounting and Quality Control in the Sage Differentially Private ML Platform

Mathias Lécuyer, Riley Spahn, Kiran Vodrahalli, Roxana Geambasu, and Daniel Hsu
Columbia University

Machine Learning (ML) introduces a dangerous double standard for data protection



- Data access through traditional code is carefully controlled.
- ML models can leak user data [4, 3, 12]. Yet they are used to make predictions for all users, shipped to everyone's devices [2, 7, 8], and sometimes shared [7, 8, 11].

Differential Privacy (DP): A rigorous tool to bound data exposure

DP randomizes a computation over a dataset (e.g. training one model) to bound the leakage of individual entries in the dataset through the output of the computation (the model). Each new DP computation increases the bound over data leakage, and can be seen as consuming part of a fixed *privacy budget*.

Formally, a randomized algorithm $\mathcal{A} : \mathcal{D} \rightarrow \mathcal{Y}$ is (ϵ, δ)-DP if for any datasets d, d' differing in one record, and for any $\mathcal{S} \subseteq \mathcal{Y}$:

$$P(\mathcal{A}(d) \in \mathcal{S}) \leq e^{\epsilon} P(\mathcal{A}(d') \in \mathcal{S}) + \delta.$$

The rich DP literature provides:

- A wealth of DP ML training algorithms (e.g., [1, 9, 10, 5]).
- Empirical [4, 12] and theoretical [6] evidence that DP prevents leaks from ML models.

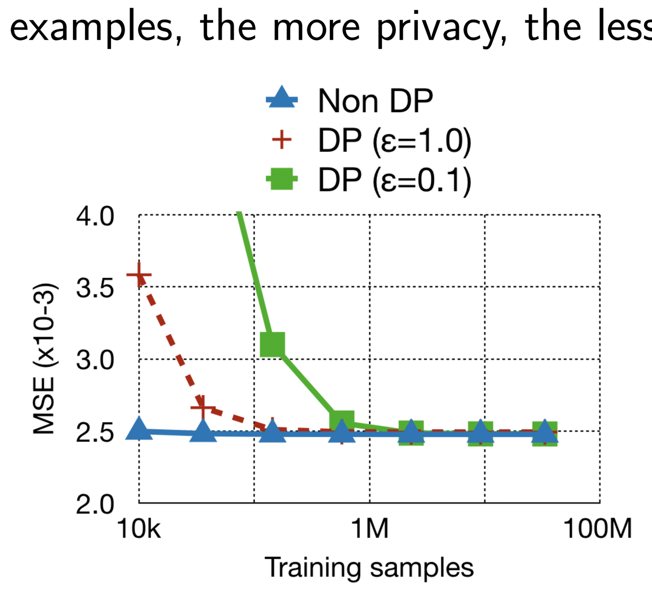
Two practical challenges of DP in ML applications

Challenge 1: running out of privacy budget. DP is typically studied in two settings:

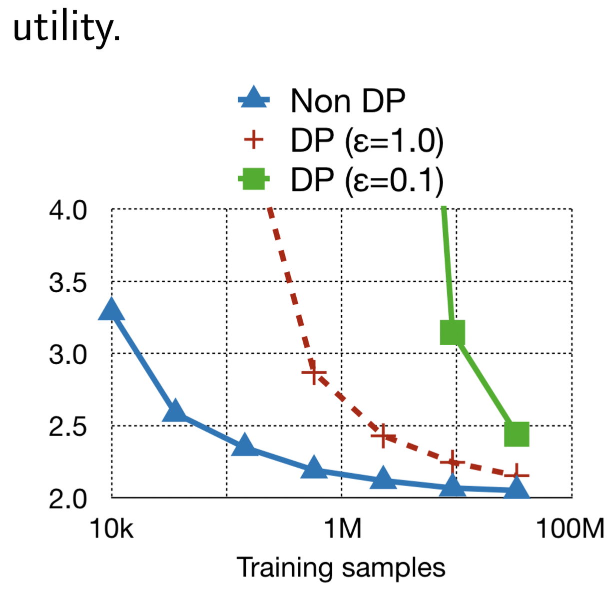
- The static database model, where no new data is ever added.
- The streaming model, where all updates are online and old data is never revisited.

In each case, an ML application will run out of privacy budget or data.

Challenge 2: the privacy/utility tradeoff. DP adds noise to ML training algorithms, reducing utility. For a fixed number of training examples, the more privacy, the less utility.



Linear regression

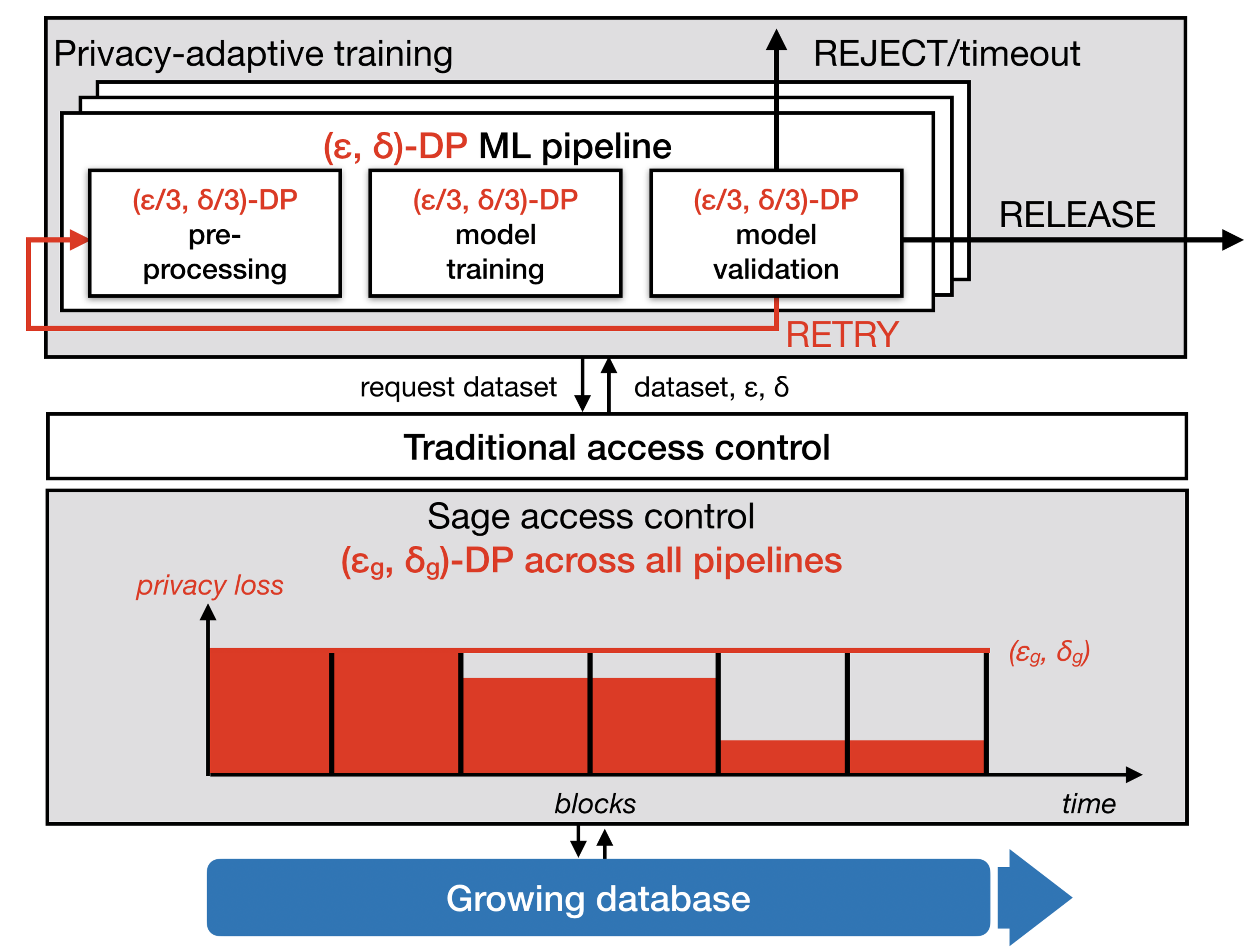


Deep neural network

Can we make Differential Privacy practical for ML applications?

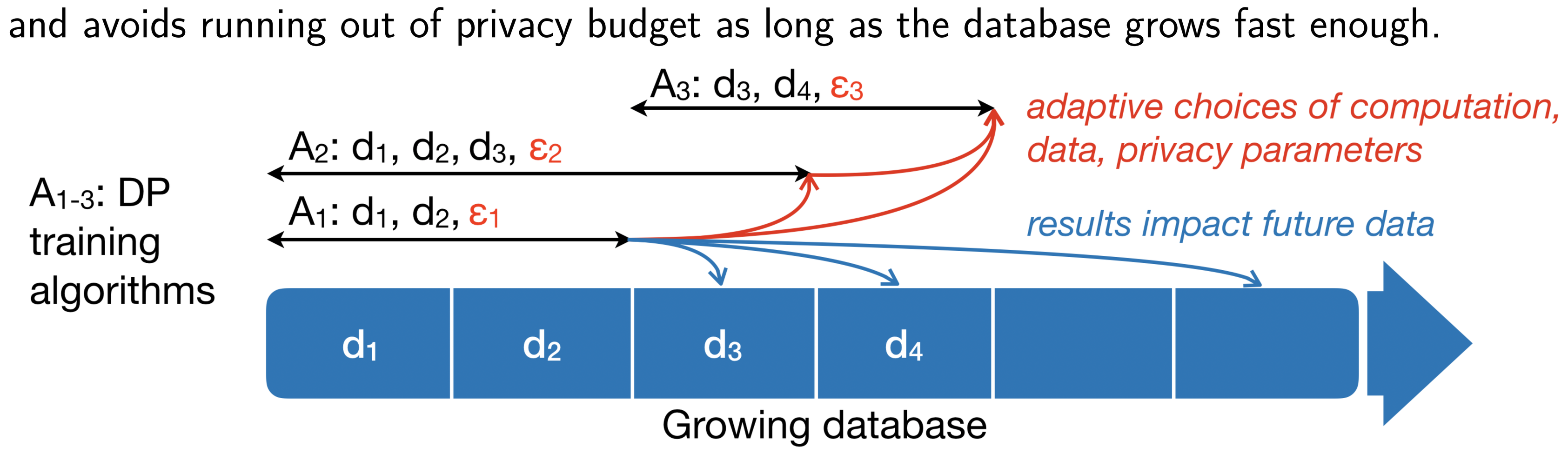
- ML workloads consist in many algorithms operating on *growing databases*.
- Algorithms both incorporate new data and reuse old data, often times adaptively.
- Sage leverages this *adaptive reuse of old data coupled with new data* to address the preceding two challenges.

The Sage architecture



Block composition to avoid running out of privacy budget

Sage introduces block composition, a new privacy accounting method that both allows efficient training on growing databases and avoids running out of privacy budget as long as the database grows fast enough.



Theorem. The ML application's total privacy loss is upper-bounded by the maximum privacy loss of any block:

$$|\text{PrivacyLoss}(\text{Growing Database})| \leq \max_k |\text{PrivacyLoss}(d_k)|.$$

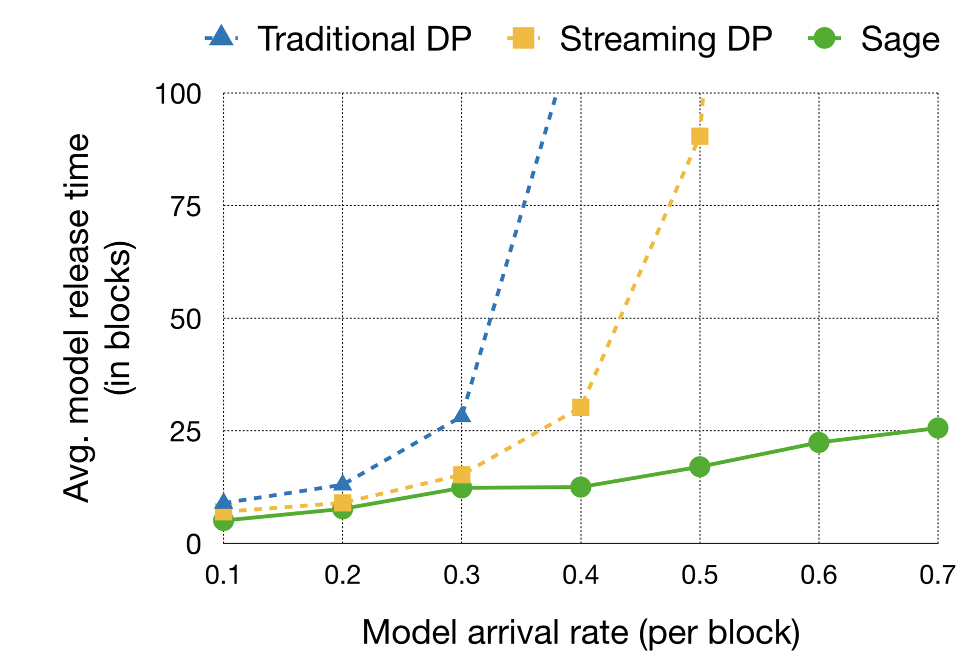
Sage's access control leverages block composition to enforce an (ϵ_g, δ_g)-DP guarantee over all models released by the application. Data blocks with exhausted budget are retired, while new data is used to train new models.

Privacy-adaptive training to control the privacy/utility trade-off

Privacy-adaptive training relies on two mechanisms to release high quality models with high probability:

1. A statistical test of model quality that is DP, and accounts for DP noise to give reliable results.
2. An iterative method that retrains models on increasing privacy budgets/data sizes until the model meets programmer-specified quality criteria.

End-to-end performance of Sage



Summary and future work

1. DP literature focused on individual ML algorithms, on static databases (no new data) or online streaming (single use data).
2. ML workloads operate on *growing databases*: models incorporate new data and (adaptively) reuse old data.
3. Sage is the first to adapt DP theory and practice to ML workloads on growing databases, for data protection.

Using DP for data protection introduces a **new global resource: the privacy budget**. Identifying principled approaches to allocate this resource is an open problem that systems researchers are uniquely positioned to address.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Conference on Computer and Communications Security (CCS)*, 2016.

[2] D. Baylor, E. Breck, H.-T. Cheng, N. Fiedel, C. Y. Foo, Z. Haque, S. Haykal, M. Ispir, V. Jain, L. Koc, C. Y. Koo, L. Lew, C. Mewald, A. N. Modi, N. Polyzotis, S. Ramesh, S. Roy, S. E. Whang, M. Wicke, J. Wilkiewicz, X. Zhang, and M. Zinkevich. TFX: A Tensorflow-based production-scale machine learning platform. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2017.

[3] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. "You Might Also Like:" Privacy risks of collaborative filtering. In *Symposium on Security and Privacy (S&P)*, 2011.

[4] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, 2019.

[5] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 2011.

[6] C. Dwork, A. Smith, T. Steinke, and J. Ullman. Exposed! A survey of attacks on private data. *Annual Review of Statistics and Its Application*, 2017.

[7] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, J. Law, K. Lee, J. Lu, P. Noordhuis, M. Smelyanskiy, L. Xiong, and X. Wang. Applied machine learning at Facebook: A datacenter infrastructure perspective. In *International Symposium on High-Performance Computer Architecture (HPCA)*, 2018.

[8] L. E. Li, E. Chen, J. Hermann, P. Zhang, and L. Wang. Scaling machine learning as a service. In *International Conference on Predictive Applications and APIs*, 2017.

[9] H. B. McMahan and G. Andrew. A general approach to adding differential privacy to iterative training procedures. *arXiv*, 2018.

[10] F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the Netflix prize contenders. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

[11] D. Shiebler and A. Tayal. Making machine learning easy with embeddings. *Systems and Machine Learning (SysML)*, 2010.

[12] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *Symposium on Security and Privacy (S&P)*, 2017.