Privacy and machine learning: two unexpected allies?

Apr 29, 2018

by Nicolas Papernot and Ian Goodfellow

In many applications of machine learning, such as machine learning for medical diagnosis, we would like to have machine learning algorithms that do not memorize sensitive information about the training set, such as the specific medical histories of individual patients. *Differential privacy* is a framework for measuring the privacy guarantees provided by an algorithm. Through the lens of differential privacy, we can design machine learning algorithms that responsibly train models on private data. Our works (with Martín Abadi, Úlfar Erlingsson, Ilya Mironov, Ananth Raghunathan, Shuang Song and Kunal Talwar) on differential privacy for machine learning have made it very easy for machine learning researchers to contribute to privacy research—even without being an expert on the mathematics of differential privacy. In this blog post, we'll show you how to do it.

The key is a family of algorithms called *Private Aggregation of Teacher Ensembles* (PATE). One of the great things about the PATE framework, besides its name, is that anyone who knows how to train a supervised ML model (such as a neural net) can now contribute to research on differential privacy for machine learning. The PATE framework achieves private learning by carefully coordinating the activity of several different ML models. As long as you follow the procedure specified by the PATE framework, the overall resulting model will have measurable privacy guarantees. Each of the individual ML models is trained with ordinary supervised learning techniques, which many of our readers are probably familiar with from hacking on ImageNet classification or many other more traditional ML pursuits.

If anyone can design a better architecture or better training algorithm for any of the individual models used by PATE, then they can also improve supervised learning itself (that is, non-private classification). Indeed, differential privacy can be thought of as a regularizer capable of addressing some of the problems commonly encountered by practitioners—even in settings where privacy is not a requirement. This includes overfitting. We elaborate in this post on these pleasant synergies between privacy and learning. In particular, we present a recent extension of PATE that refines how the different ML models are coordinated to simultaneously improve both the accuracy and privacy of the model resulting from the PATE framework. This shows how aligned the goals of differential privacy are with the pursuit of learning models that generalize well.

Why do we need private machine learning algorithms?

Machine learning algorithms work by studying a lot of data and updating their parameters to encode the relationships in that data. Ideally, we would like the parameters of these machine learning models to encode general patterns (“patients who smoke are more likely to have heart disease”) rather than facts about specific training examples (“Jane Smith has heart disease”). Unfortunately, machine learning algorithms do not learn to ignore these specifics by default. If we want to use machine learning to solve an important task, like making a cancer diagnosis model, then when we publish that machine learning model (for example, by making an open source cancer diagnosis model for doctors all over the world to use) we might also inadvertently reveal information about the training set. A malicious attacker might be able to inspect the published model and learn private information about Jane Smith [SSS17]. This is where differential privacy comes in.

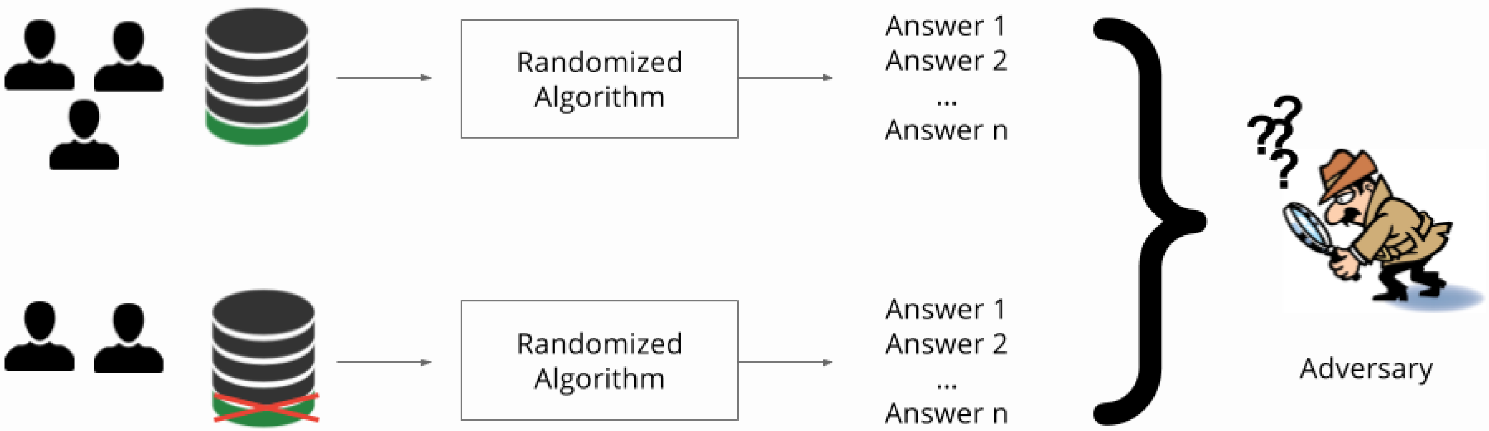
How can we define and guarantee privacy?

Scientists have proposed many approaches to provide privacy when analyzing data. For instance, it is popular to anonymize the data before it is analyzed, by removing private details or replacing them with random values. Common examples of details often anonymized include phone numbers and zip codes. However, anonymizing data is not always sufficient and the privacy it provides quickly degrades as adversaries obtain auxiliary information about the individuals represented in the dataset. Famously, this strategy allowed researchers to de-anonymize part of a movie ratings dataset released to participants of the Netflix Prize when the individuals had also shared their movie ratings publicly on the Internet Movie Database (IMDb) [NS08]. If Jane Smith had assigned the same ratings to movies A, B and C in the Netflix Prize dataset and publicly on IMDb, then researchers could link data corresponding to Jane across both datasets. This would in turn give them the means to recover ratings that were included in the Netflix Prize but not on IMDb. This example shows how difficult it is to define and guarantee privacy because it is hard to estimate the scope of knowledge—about individuals—available to adversaries.

Differential privacy is a framework for evaluating the guarantees provided by a mechanism that was designed to protect privacy. Invented by Cynthia Dwork, Frank McSherry, Kobbi Nissim and Adam Smith [DMNS06], it addresses a lot of the limitations of previous approaches like k-anonymity. The basic idea is to randomize part of the mechanism's behavior to provide privacy. In our case, the mechanism considered is always a learning algorithm, but the differential privacy framework can be applied to study any algorithm.

The intuition for introducing randomness to a learning algorithm is to make it hard to tell which behavioral aspects of the model defined by the learned parameters came from randomness and which came from the training data. Without randomness, we would be able to ask questions like: "What parameters does the learning algorithm choose when we train it on this specific dataset?" With randomness in the learning algorithm, we instead ask questions like: "What is the probability that the learning algorithm will choose parameters in this set of possible parameters, when we train it on this specific dataset?"

We use a version of differential privacy which requires that the probability of learning any particular set of parameters stays roughly the same if we change a single training example in the training set. This could mean to add a training example, remove a training example, or change the values within one training example. The intuition is that if a single patient (Jane Smith) does not affect the outcome of learning, then that patient's records cannot be memorized and her privacy is respected. In the rest of this post, we often refer to this probability as the *privacy budget*. Smaller privacy budgets correspond to stronger privacy guarantees.



In the above illustration, we achieve differential privacy when the adversary is not able to distinguish the answers produced by the randomized algorithm based on the data of two of the three users from the answers returned by the same algorithm based on the data of all three users.

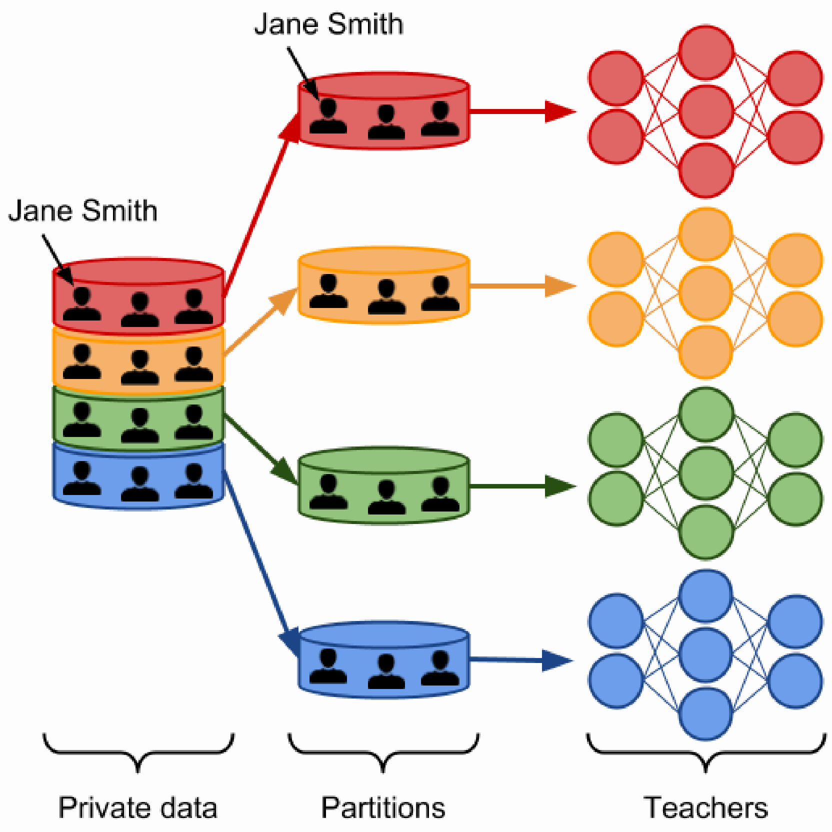
What is the intuition behind PATE?

Our PATE approach at providing differential privacy to machine learning is based on a simple intuition: if two different classifiers, trained on two different datasets with no training examples in common, agree on how to classify a new input example, then that decision does not reveal information about any single training example. The decision could have been made with or without any single training example, because both the model trained with that example and the model trained without that example reached the same conclusion.

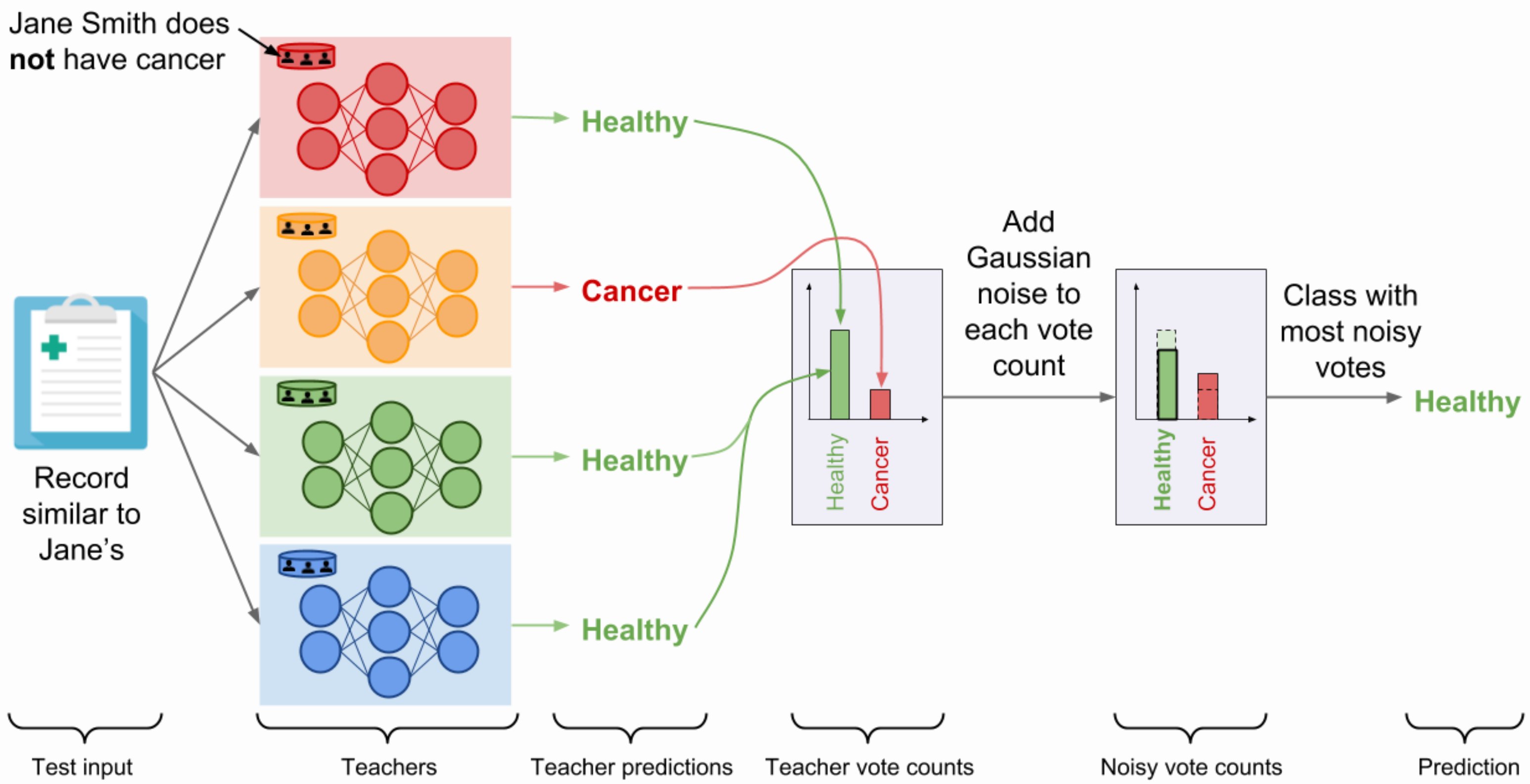
Suppose then that we have two models trained on separate data. When they agree on an input, it seems like maybe we can publish their decision. Unfortunately, when they do not agree, it is less clear what to do. We can't publish the class output by each model separately, because the class predicted by each model may leak some private information contained in its training data. For instance, assume that Jane Smith contributed to the training data of one of the two models only. If that model predicts that a patient whose record is very similar to Jane's has cancer whereas the other model predicts the contrary, this reveals private information about Jane. This simple example illustrates why adding randomness to an algorithm is a requirement to ensure it provides any meaningful privacy guarantees.

How does PATE work?

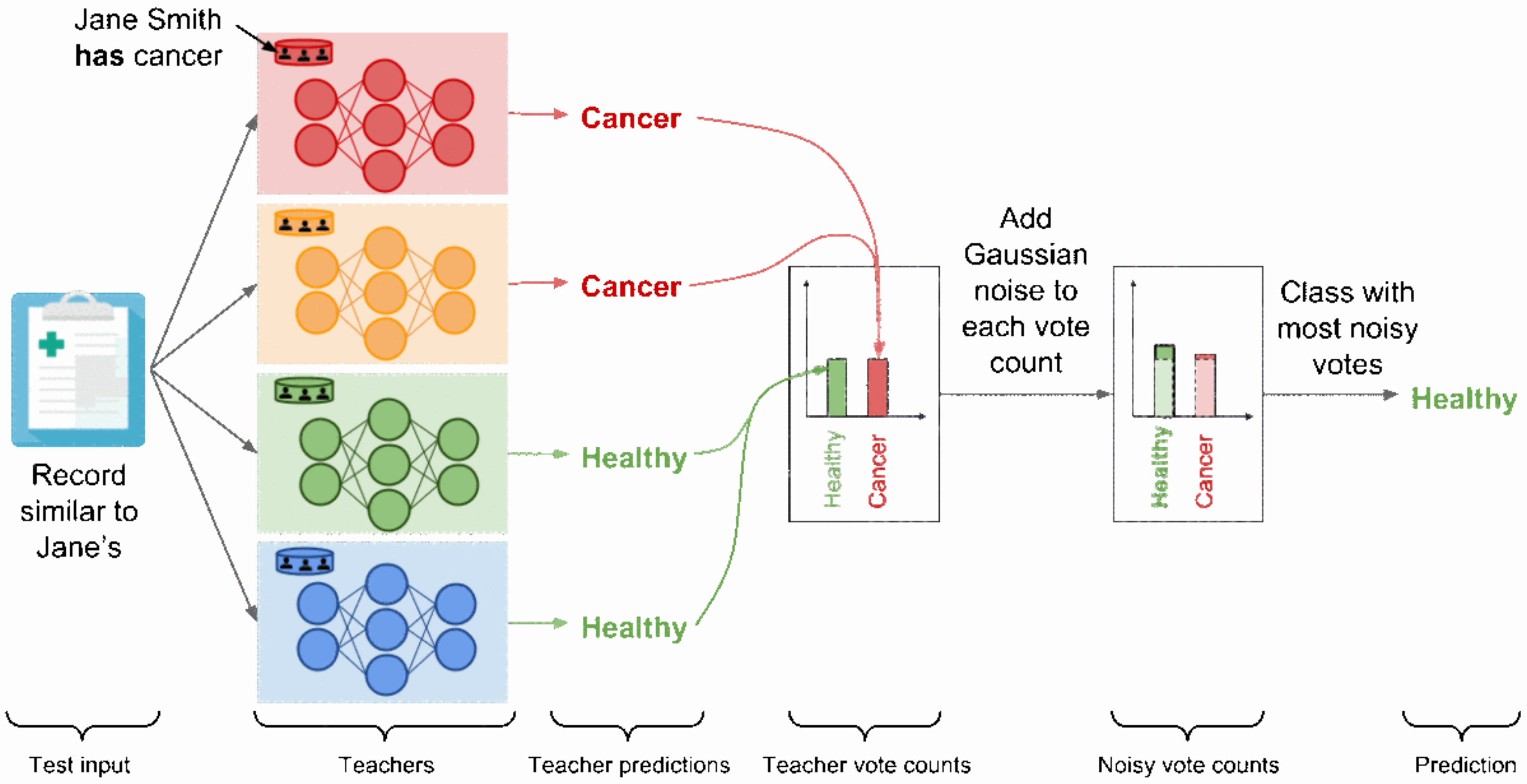
Let us now see step-by-step how the PATE framework builds on this observation to learn responsibly from private data. In PATE, we start by partitioning the private dataset in subsets of data. These subsets are partitions, so there is no overlap between the data included in any pair of partitions. If Jane Smith's record was in our private dataset, then it is included in one of the partitions only. We train a ML model, called a *teacher*, on each of these partitions. There are no constraints on how the teachers are trained. This is in fact one of the main advantages of PATE: it is agnostic to the learning algorithm used to create teacher models. All of the teachers solve the same machine learning task, but they were trained independently. That is, only one of the teachers has analyzed Jane Smith's record during training. Here is an illustration of what this part of the framework looks like.



We now have an ensemble of teacher models that were trained independently, but without any guarantees of privacy. How do we use this ensemble to make predictions that respect privacy? In PATE, we add noise while aggregating the predictions made invidually by each teacher to form a single common prediction. We count the number of teachers who voted for each class, and then perturb that count by adding random noise sampled from the Laplace or Gaussian distribution. Readers familiar with the differential privacy literature will recognize the noisymax mechanism. When two output classes receive an equal (or quasi equal) number of votes from the teachers, the noise will ensure that the class with the most number of votes will be one of these two classes chosen at random. On the other hand, if most of the teachers agreed on the same class, adding noise to the vote counts will not change the fact that this class received the most votes. This delicate orchestration provides correctness and privacy to the predictions made by the noisy aggregation mechanism—as long as the consensus among teachers is sufficiently high. The following figure depicts the aggregation mechanism is a setting where consensus among teachers is high: adding random noise to the vote counts does not change the candidate label.



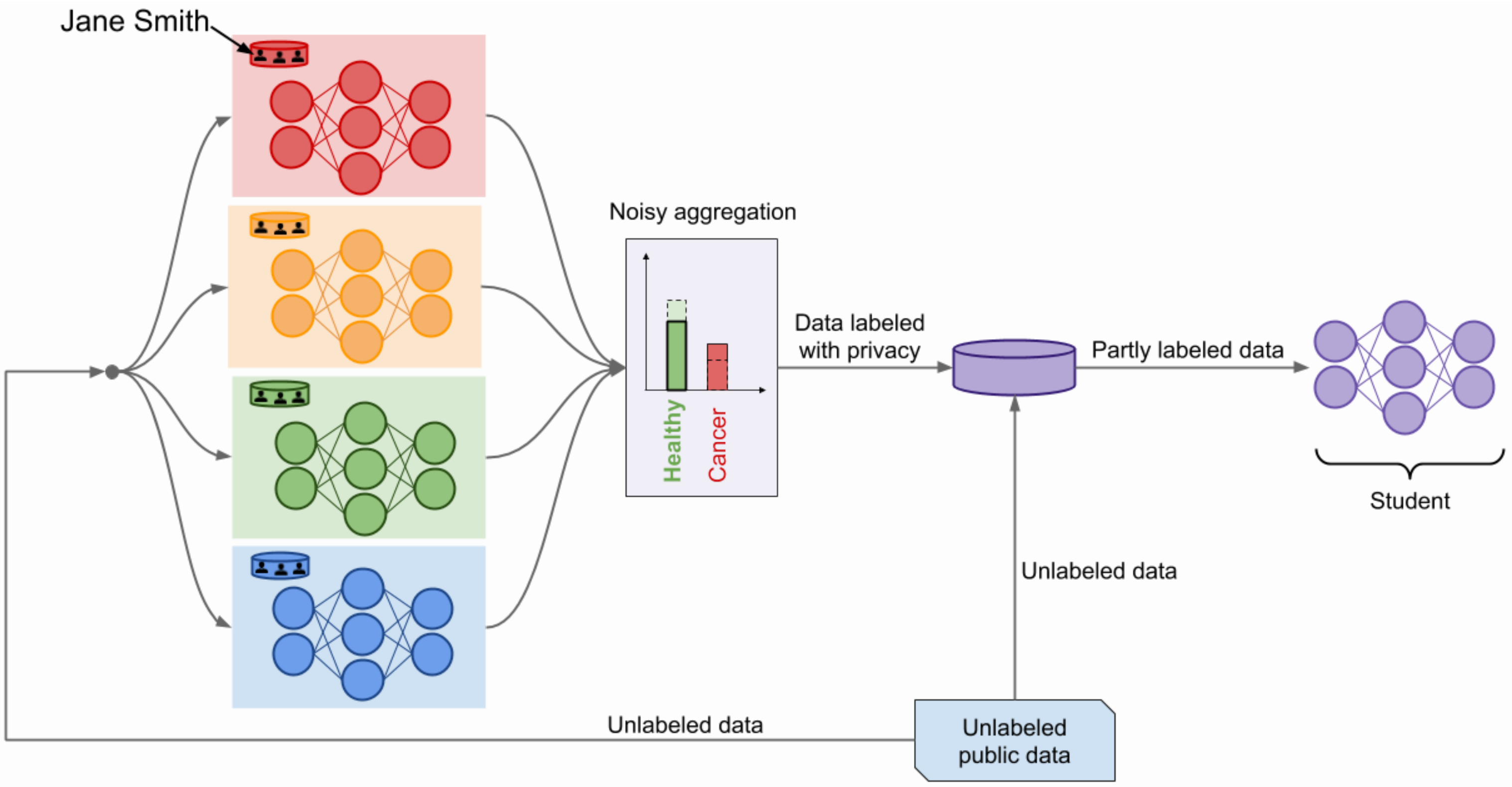
For clarity, we illustrated the aggregation with a binary medical diagnosis task but the mechanism extends to a large number of classes. Now, let's analyze the outcome of this mechanism if Jane Smith had cancer. The red model—the only teacher trained on the partition containing Jane Smith's data—has now learned that a record similar to Jane's is characteristic of a patient suffering from cancer, and as a consequence changes its prediction on the test input (which is similar to Jane's) to "Cancer". There are now two teachers voting for the label “Cancer” while the two remaining teachers vote for “Healthy”. In these settings, the random noise added to both vote counts prevents the outcome of aggregation from reflecting the votes of any individual teachers to protect privacy: the noisy aggregation’s outcome is equally likely to be “Healthy” or “Cancer”.



At this point, PATE provides what can be thought of as a differentially private API: each label predicted by the noisy aggregation mechanism comes with rigorous differential privacy guarantees that bound the privacy budget spent to label that input. In our running example, we can bound the probability that the label predicted was influenced by any of the invididual records on which we trained teachers, including Jane Smith’s for instance. We apply one of two techniques called the Moments Accountant [ACG16] and Renyi Differential Privacy [M17] to compute this bound. Using the histogram of votes for each query, we estimate the probability of the aggregation's outcome to change as a result of the noise injected. We then aggregate this information over all queries. In practice, the privacy budget primarily depends on the consensus between teachers and how much noise is added. Higher consensus between teachers, as expressed by the largest number of votes assigned to a class, tend to favor smaller privacy budgets. Up to a certain point, adding larger amounts of noise before counting votes assigned by teachers also yields smaller privacy budget. Recall that smaller privacy budgets correspond to stronger privacy guarantees.

However, the framework faces two limitations at this point. First, each prediction made by the aggregation mechanism increases the total privacy budget. This means that the total privacy budget eventually becomes too large when many labels are predicted—at which point the privacy guarantees provided become meaningless. The API would therefore have to impose a maximum number of queries across all users, and obtain a new set of data to learn a new teacher ensemble when that cap has been reached. Second, we can't publicly publish the ensemble of teacher models. Otherwise, an adversary could inspect the internal parameters of the published teachers to learn things about the private data they trained on. For these two reasons, there is one addtional step in PATE: creating a student model.

The student is trained by transfering knowledge acquired by the teacher ensemble in a privacy-preserving way. Naturally, the noisy aggregation mechanism is a crucial tool for this. The student selects inputs from a set of unlabeled public data and submits these inputs to the teacher ensemble to have them labeled. The noisy aggregation mechanism responds with private labels, which the student uses to train a model. In our work, we experimented with two variants: PATE trains the student only on labeled inputs (in a supervised fashion) whereas PATE-G trains the student on both labeled and unlabeled inputs (in a semi-supervised fashion with Generative Adversarial Networks or Virtual Adversarial Training).



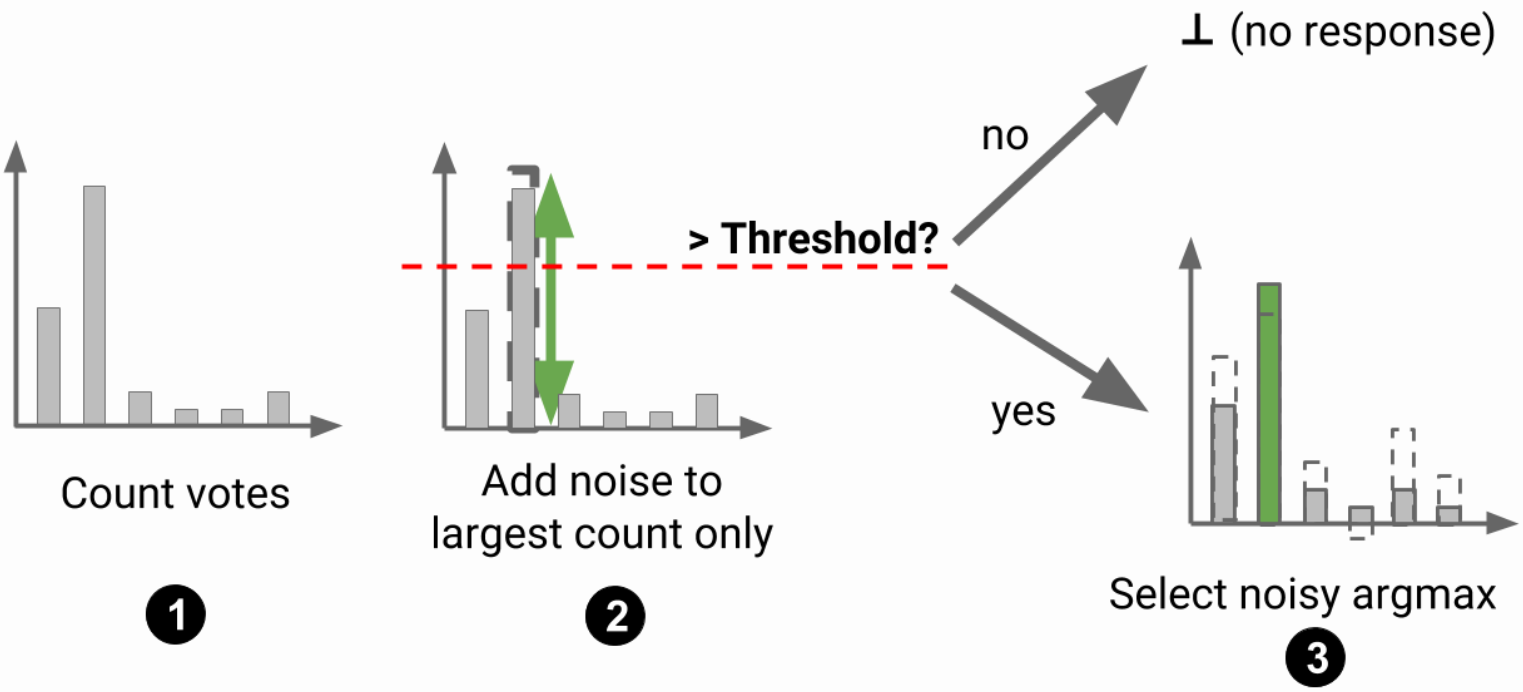
The student model is the final product of PATE. It is deployed to respond to any prediction queries from end users. At this point, the private data and teacher models can safely be discarded: the student is the only model used for inference. Observe how both pitfalls identified above are now addressed. First, the overall privacy budget is now fixed to a constant value once the student has completed its training. Second, an adversary with access to the student's internal parameters could in the worst-case only recover the labels on which the student trained, which are private. This guarantee stems from the noisy aggregation mechanism.

Pleasant synergies between privacy and learning with PATE

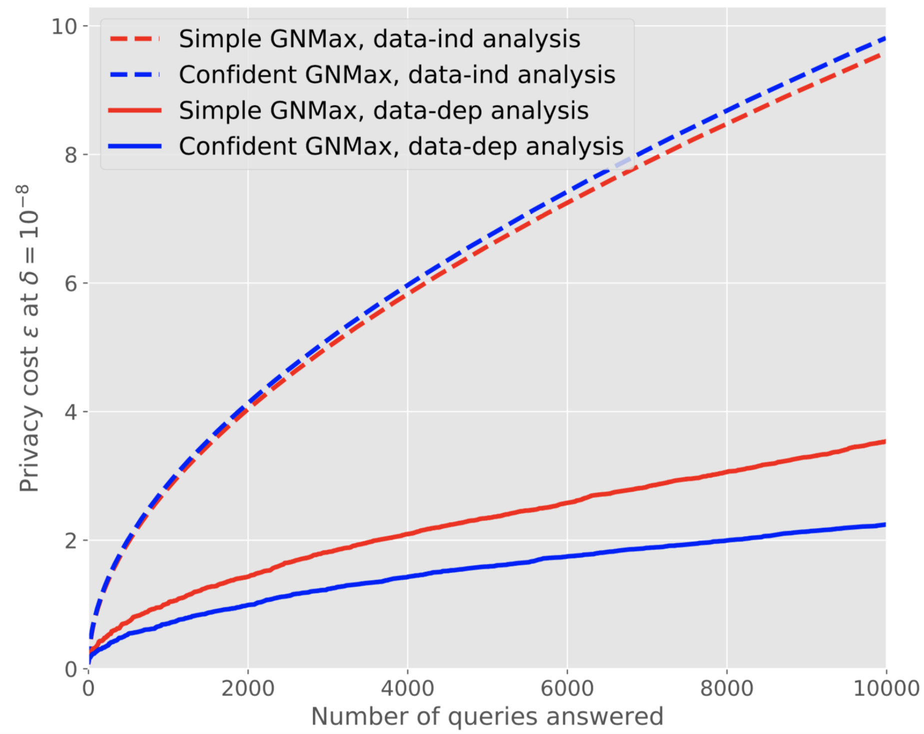
You may have noticed that privacy guarantees and the correctness of a label predicted by the aggregation mechanism stem from a strong consensus between teachers. Indeed, when most of the teachers agree on a prediction, it is unlikely that adding noise will modify the class that received the largest number of teacher votes. This results in a very strong privacy guarantee for the aggregation mechanism. Similarly, many teachers agreeing on a label indicates confidence in the correctness of that prediction because these teacher models were trained independently from different data partitions. This is intuitively why PATE is capable of exploiting some pleasant synergies between privacy and learning.

This can be surprising. Indeed, it is common to present differential privacy as a property that is nice-to-have but creates a necessary trade-off with performance. However, things are different with machine learning. Differential privacy is in fact well aligned with the goals of machine learning. For instance, memorizing a particular training point—like the medical record of Jane Smith—during learning is a violation of privacy. It is also a form of overfitting and harms the model's generalization performance for patients whose medical records are similar to Jane's. Moreover, differential privacy implies some form of stability (but the opposite is not necessarily true).

This observation motivated the design of a refined aggregation mechanism for PATE in our recent paper following up with the original work. This new mechanism—the Confident Aggregator—is *selective*: teachers respond to only some of the queries made by the student. When a query is made by the teacher, we first check whether the consensus among teachers is sufficiently high. If the number of votes assigned to the label most popular among teachers is larger than a threshold, we accept the student's query. If not, we reject it. The threshold itself is randomized in order to provide privacy during this selection process. Once a query has been selected, we proceed with the original noisy aggregation mechanism: we add noise to each of the vote counts corresponding to each label and return the label with the most votes. This procedure is illustrated below (on a task with 6 classes to avoid misleading simplifications to the figure in the binary case).



In practice, this means that our privacy budget is now spent on two things: selecting and answering queries. However, because the queries we elect to answer are characterized by a high consensus between teachers, the budget needed to answer them is very small. In other words, we can think of the Confident Aggregator as a mechanism that filters out the queries that would consume most of our privacy budget in the original mechanism. As a result, the total privacy budget afforded by the Confident Aggregator is smaller than that of the original noisy aggregation mechanism at comparable levels of student performance. The graph below visualizes this improvement as a function of the number of (student) queries answered by the original mechanism (Simple GNMax) and refined mechanism (Confident GNMax) when one uses a data-dependent (data-dep) analysis, which we do by applying the Moments Accountant or Renyi Differential Privacy.



How can ML researchers make improved models for PATE?

Two factors primarily impact the strength of privacy guarantees provided by our approach:

1. the **consensus among teachers**: when this consensus is strong, meaning almost all teachers make the same label prediction, the privacy budget spent when outputting the corresponding label is reduced. This intuitively corresponds to a scenario where the prediction made is a generality learned by all teachers even though they were trained on disjoint sets of data.
2. the **number of student queries**: each time the student makes a label query to the teachers during its training, the budget spent by the teachers to produce this label is added to the total privacy cost. Therefore, training the student with as few teacher queries as possible reinforces the privacy guarantees provided.

Both of these points can be addressed from a purely ML perspective. Strengthening the teacher consensus requires that one can train many teachers with little data for each of them. Improving the individual accuracy and generalization of these models will most likely contribute to improving the consensus. Unlike teacher training, which is fully supervised, reducing the number of student queries is a semi-supervised learning problem. For instance, state-of-the-art privacy-preserving models on MNIST and SVHN were trained with PATE-G, a variant of the framework that uses Generative Adversarial Networks to train the student in a semi-supervised way. The student has access to a relatively large set of unlabeled inputs and it must learn with as little supervision from the teachers as possible.

To help spur these efforts, the PATE framework is open-sourced and available as part of the TensorFlow models repository. For the purpose of keeping things simple, the code uses publicly-available image classification datasets like MNIST and SVHN. You can clone it and set the `PYTHONPATH` variable appropriately on a UNIX machine as follows:

```
cd
git clone https://github.com/tensorflow/models
cd models
export PYTHONPATH=$(pwd):$PYTHONPATH
cd research/differential_privacy/multiple_teachers
```

The first step in PATE is then to train the teacher models. In this demo, we use the MNIST dataset and an ensemble of 250 teachers (see the PATE papers for a discussion of why that is a good choice).

```
python train_teachers.py --nb_teachers=250 --teacher_id=0 --dataset=mnist
python train_teachers.py --nb_teachers=250 --teacher_id=1 --dataset=mnist
...
python train_teachers.py --nb_teachers=250 --teacher_id=248 --dataset=mnist
python train_teachers.py --nb_teachers=250 --teacher_id=249 --dataset=mnist
```

This will save checkpoints for the 250 teachers. Now, we can load these teachers and apply the aggregation mechanism to supervise student training.

```
python train_student.py --nb_teachers=250 --dataset=mnist --stdnt_share=1000 --lap_s
```

This will train the student using the first 1000 inputs from the test set labeled using our 250 teachers and an aggregation mechanism introducing noise of laplacian scale `1/20`. This will also save a file `/tmp/mnist_250_student_clean_votes_lap_20.npy` containing all of the labels produced by the teachers, which we use to evaluate how private our student is.

To know the value of the differential privacy bounds guaranteed by our student model, we need to run the analysis script. This will perform the privacy analysis using information about the teacher consensus saved while training the student. Here, the `noise_eps` parameter should be set to `2/lap_scale`.

```
python analysis.py --counts_file=/tmp/mnist_250_student_clean_votes_lap_20.npy --max
```

This setup reproduces the PATE framework with the original noisy aggregation mechanism. Readers interested in using the Confident Agggregator mechanism introduced in our more recent paper can find the relevant code here.

More PATE ressources

- The original PATE paper at ICLR 2017 and recording of the ICLR oral
- The ICLR 2018 paper on scaling PATE to large number of classes and imbalanced data.
- GitHub code repo for PATE
- GitHub code repo for the refined privacy analysis of PATE

Conclusion

Privacy can be thought of as an ally rather than a foe in the context of machine learning. As the techniques improve, differential privacy is likely to serve as an effective regularizer that produces better-behaved models. Within the framework of PATE, machine learning researchers can also make significant contributions towards improving differential privacy guarantees without being an expert in the formal analyses behind these guarantees.

Acknowledgements

Thank you to our collaborators for their contributions to the material presented in this post. A special thank you to Ilya Mironov and Úlfar Erlingsson for their feedback on a draft of this post.

References

[ACG16] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308-318). ACM.

[DMNS06] Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006, March). Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography Conference (pp. 265-284). Springer, Berlin, Heidelberg.

[M17] Mironov, I. (2017, August). Renyi differential privacy. In Computer Security Foundations Symposium (CSF), 2017 IEEE 30th (pp. 263-275). IEEE.

[NS08] Narayanan, A., & Shmatikov, V. (2008, May). Robust de-anonymization of large sparse datasets. In Security and Privacy, 2008. SP 2008. IEEE Symposium on (pp. 111-125). IEEE.

[SSS17] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017, May). Membership inference attacks against machine learning models. In Security and Privacy (SP), 2017 IEEE Symposium on (pp. 3-18). IEEE.

cleverhans-blog

cleverhans-blog

Jekyll blog associated with cleverhans