CCS 2019 Talk Summaries

Sebastian Meiser, Visa Research

November 11, 2019

This document presents a few summaries of talks I attended at CCS 2019. I don't guarantee that the summaries are comprehensive or that they capture all subtleties of the presented matter; if you find any technical inaccuracies, please contact me about them.

1 Pre-Conference Workshop: TPDP

1.1 Encode, Shuffle, and Analyze (ESA) Revisited: Strong Privacy despite High-Epsilon

Speaker: Abhradeep Guha Thakurta, Google Research, UC Santa Cruz

Assume we have a number of devices use an anonymizer and local DP to achieve some central differential privacy. The anonymizer can either use summation or shuffling to achieve this goal. Naturally we focus on the second part here.

What we do is we take the locally DP objects, remove identifiers (if there are any), shuffle them, and release them. We want to start with weak local DP, i.e., with a high epsilon ($\varepsilon > 1$) and still achieve strong central differential privacy: We get a boost of about $\frac{1}{\sqrt{n}}$ for ε.

To this end, we look at three ideas:

- **Attribute fragmenting:** We split one-hot vectors into the separate bits, then shuffle them, and get some utility in $\Theta\left(\sqrt{\frac{\log k}{ne^{\varepsilon_{\text{local}}}}}\right)$. Note that if we have t records and a local $\varepsilon_{\text{local}} = 1$ we get local DP of $t \cdot \varepsilon_{\text{local}}$.
- **Record fragmenting:** I'm not quite sure what exactly happened here; I think the result is that instead of an ok central DP ($\varepsilon = 1.5$) trade-off that comes with a horrible local DP guarantee ($eps \approx 25$), we can have a local DP of $\varepsilon = 1$ and still get central DP with $\varepsilon = 1.5$. This degrades the utility, but Abhradeep assures us that it's not that bad.
- **Crowds:** We can group records to achieve a better local DP / utility trade-off. We split our data into crowds and analyze them. A cute idea here is to add Laplace noise Lap $\left(\frac{1}{\varepsilon_{\text{shuffle}}}\right)$ (and subtract a large enough

constant) to the count of records it has and then drop as many records as required to meet the count. If we still come up with a number higher than the actual count, we have a distinguishing event.

1.2 DPella

Speaker: Elisabet Lobo Vesga

We know how to do queries with DP and how to estimate the accuracy. However, what happens to our accuracy if we want to add and combine the results of several queries?

In comes DPella, a Haskell library, which allows us to keep track the privacy and accuracy of the (combined) queries we ask and to find out the privacy budget used by a program (via symbolic execution). Similarly, we can get an estimate of the accuracy of the program. That sounds pretty interesting.

So far, they only consider the Laplace mechanism, but they are working on integrating the Gauss mechanism as well.

1.3 Private Stochastic Convex Optimization with Optimal Rate

Speaker: Abhradeep Guha Thakurta, Google Research, UC Santa Cruz

When we design a (differentially private) learning algorithm, we have to consider population risk. Here we have a convex loss function, which makes things easier in many ways. The excess population risk is the expected loss on a random sample, when compared with the minimal such loss. $\max\left(\frac{1}{\sqrt{n}}, \frac{\sqrt{d}}{\varepsilon n}\right)$, the first part of which is what we get for the non-private case and it turns out we often still achieve that.

We look at a noisy SGD with a batch size of approximately $\max\left(\sqrt{n}, \sqrt{d}\right)$; the number of rounds we need is about $\min\left(n, \frac{n^2}{d}\right)$.

1.4 Lessons learned from the NIST DP Synthetic Data Competition

Speaker: Ryan McKenna, University of Massachusetts, Amherst

The talk is about differentially private synthetic data, which is very interesting for a number of reasons. If we have private synthetic data, we can use arbitrary mechanisms on them and we can use it however we want without needing to keep track of any budget.

The speaker discusses the competition he partook in. The data was US census data with 98 attributes and 661k individuals. All the data fields was made of integers between 0 and a known maximum. The synthetic data was judged on: accuracy on all 3-way marginals (10^13 queries) and an approximation of high-order conjunctions (such as ”how many records have an age in range X and income in range Y?”).

The way they approached this was to first compute a range of noisy aggregate statistics and to then use an inference engine to create synthetic data from it.

1. Measure 1-way marginals using the Gaussian mechanism, treat counts below a threshold as zero.
2. Construct a correlation graph between any pairs of attributes. Each attribute is a node in the graph and the edge weights correspond to the correlation between the attributes. They construct this on the provisional dataset that is assumed to be public (we need to sacrifice the privacy of this dataset). We find a maximum spanning tree of the graph (of the provisional data), then measure 2-way marginals in a differentially private way on the actual data for each part of the tree.

The top 4 solutions of the competition were fairly similar, with the winning one (the one presented) being unique in terms of the inference engine used. Their mechanism ran in 30 minutes.

1.5 Full Convergence of the iterative Bayesian update and applications to local differential privacy

Speaker: Catuscia Palamidessi

In the local DP setting, every user can theoretically set their own privacy level and they don't need to be the same over all users. In this talk we focus on the statistical utility of the data and we try to retrieve the original distribution of the data.

Catuscia presents the iterative Bayesian update: they start with any distribution with full support, e.g., uniform, and then update the distribution iteratively. This approach achieves a pretty good approximation of the original distribution; the work presented fixes a bug of this approach, extends it to more mechanisms, and compares the technique to other inversion techniques.

1.6 Differentially private real summation with single- and multi-message shuffling

Speaker: James Bell

This work too talks about the shuffle model, here using the randomized response mechanism. In their analysis, they allow the adversary to know whether each participant (other than the one of interest) lied or told the truth. Each participant that lied naturally introduces some noise that helps protect the one remaining real entry. Thus, this technique relies on at least a certain number of other participants to add noise correctly in order to achieve privacy.

The work(s) also look at reducing the amount of communication necessary to achieve a good trade-off when more messages per party are allowed. They reduce the number of required messages from about $\sqrt{n}$ to $\log(n)$.

1.7 Differentially private release of synthetic graphs

Speaker: Marek Elias

We look at social networks and Marek starts off the talk with the example of twitter, more precisely, re-tweets between hardcore democrats and republicans.

To achieve privacy, we want to start with a graph G and create a differentially private graph G' and we want to preserve the weight of cuts. Existing work by Gupta et al. creates a fully connected graph with Laplace noise on the edges. That's okay if we have a graph with lots of edges already, but for the sparse graphs we're looking at, it would actually be more correct to output an empty graph.

Here we use a sparsifier that preserves cut sizes with a small multiplicative error, however, exponential time is required. This work provides the first non-trivial guarantee that can be computed in polynomial time. I'm not sure how good / useful this actually is in practice.

1.8 Privacy hypothesis testing via robustness

Speaker: Audra McMillan

Audra focuses on how to design private testing algorithms. We start with yes-or-no questions and generate hypotheses out of them.

Here, a test gets a database as an input distinguishes between the null hypothesis and an alternative hypothesis, i.e., it outputs a bit. Differential privacy is useful here not just for privacy, because it provides stability and that's a valuable property to have anyway.

We look at two different problems here:

- in simple hypothesis testing we try to compare two different distributions. That's very much standard DP stuff.

  In a non-private way, we solve this problem as follows: given a dataset X, we output P if it's more likely than Q and Q otherwise.

  We can rewrite this as follows:

  $$L(X) = \sum_{x \in X} \log \frac{P(x)}{Q(x)}$$

  Now we can replace the test by:

  $$LLR(X) = P \text{ if } L(X) \geq 0, Q \text{ if } L(x) < 0$$

  This is pretty similar to an operation on the privacy loss. We seem to get some privacy for free here by making the tests more robust.

- Alternatively we can look at identity testing in high dimensions. Here, we basically compare a uniform distribution from a product distribution that's far from the uniform distribution. Here a main insight is that the global sensitivity is extremely large with a high number of dimensions. However, the worst-cases that actually give us such worst-cases are really rare and don't at all look uniform, so we should be able to distinguish them easier anyway.

They define a set of "good datasets" coming with a test that is insensitive and that rejects datasets that aren't in the set of "good datasets". As a first step, this test is used to reject things that are obviously non-uniform.

Fortunately there is a function $\hat{T}$ that has a sensitivity of T on the good dataset and that satisfies $\hat{T}(X) = T(X)$ in that region. Basically, $\hat{T}$ is like the test we want to use, but insensitive anywhere. We then use $\hat{T}(X) + \text{Lap}\left(\frac{\lambda}{\varepsilon}\right)$

If we sample from the uniform distribution then any two samples should be independent, i.e., their inner products should be small. Consequently we define the good region as datasets with small inner products and that don't have a bias in an individual dimension that is too high.

1.9 Private hypothesis selection

Speaker: Mark Bun

Given a publicly known collection of distributions H and some i.i.d. samples $x_1, \ldots, x_n$ from some unknown distribution P and we want to find a hypothesis $h \in H$ that is close to P in total variation distance: If there is a hypothesis $h^* \in H$ s.t. $\text{TV}(P, h^*) \leq \alpha$ then with high probability we output some $h \in H$ with $\text{TV}(P, h) \leq O(\alpha)$.

This work provides a robust variant of hypothesis testing, gives sample-efficient algorithms fir distribution learning and can be used as pre-processing for other applications.

Why use total variation distance? It's mathematically convenient and it's insensitive to low probability events.

In a non-private way we can achieve this with a Scheffe Tournament that lets the hypotheses compete against each other. If one hypothesis h_1 wins against another $h2$, then the winner is at most some small error term away from the $\min\{TV(h_1, P), TV(h_2, P)\}$.

To achieve differential privacy, we add noise to the tests (here, Laplace noise). If the number of samples is high enough, we can achieve privacy somewhat easily, but we'd like to avoid this large number of samples.

This work uses the exponential mechanism and encodes something smart as the quality function q of the exponential mechanism. We can't use the number of contests won, as the sensitivity of that would be really high. Mark introduces a slightly modified contest that introduces draws to the picture and then sets q as the minimum number of samples that have to be changed for h to lose at least one contest.

1.10 The search for anonymous data: attacks against privacy preserving methods and systems

Speaker: Yves-Alexandre de Montjoye

Yves starts his talk by making clear that this is an interactive session. The first question he asks is how we can find a balance between data (which is very useful and enables great utility, e.g., via machine learning) and privacy.

In practice, people try to preserve privacy by providing anonymity of the data, which fits well to both the GDPR (in Europe) and the CCPA (in California). Yves highlights that this is problematic, since large datasets of released anonymized information were de-anonymized later on by researchers.

First Act: Pseudonymization Yves holds the opinion that mere pseudonymization does is not sufficient to provide privacy. From a privacy-perspective, that is certainly true.

What does it take to identify someone in a dataset? As one example, he shows that in a mobile phone dataset 4 points of hour + location are enough to uniquely identify 95% of the users.

Second Act: De-identification Here, we try to make it harder for the attacker to find data points, e.g., by adding noise. For the mobile phone dataset, they added a significant amount of noise to every data point (up to several kilometers). We have that the uncertainty $\sim (v \cdot h)^{\frac{-p}{100}}$, where h is the soatial resolution, v the temporal resolution and p the points known to the attacker.

Yves then discusses the argument "you can never be sure" (that the person you think you found even was in the dataset). The question here is, whether or not we can quantify this uncertainty. Basically: how likely are we to have found the actual record?

To answer this question, they took a small subset of the dataset (way less than 1%), learned to re-identify records and then test whether they can predict if a record s=is unique or not, which they evaluate on the whole dataset. [1]

Yves confludes this acy by saying that anonymization (in the traditional, de-identification sense) doesn't work for high dimensional data.

Diffix, a heuristic query-based system Diffix is a privacy preserving system, (although it doesn't provide differential privacy). The way it works is that every query is answered with a combination of static and dynamic noise. For every condition of the query (say: the number of people with age=40, dept=Computing, high-salary=True), they have a static noise term and they also add a dynamic noise term each. They hoped to achieve (virtually) unlimited queries with somewhat little noise.

Yves and his group developed an attack on this system.

The assumptions are that there is only one target and the attacker wants to infer a specific attribute of Bob (binary).

[1] their model is available here: `cgp.doc.ic.ac.uk/individual-risk`

They send two queries, one with one less condition. Now the static noise terms of the same conditions remain the same and only one noise term (of the condition that was changed) as well as the dynamic noise terms remain. Now, if Bob, say, has a high-salary and if they can assume that Bob is also unique in these attributes, they get two different distributions. They then exploit the background knowledge of the attacker, use a simple likelihood-test and can then infer Bob's attribute with high certainty, depending on the number of known attributes of Bob (and on the dataset); for most datasets, the attacker needs background knowledge about between 10 and 30 attributes.

Diffix proposed a patch that seems to mostly target the specific attack (by disallowing "dangerous queries"), but that does not seem to fix the underlying issue.