Lecture 19: Additional topics in private machine learning

December 5, 2019

1 Efficient algorithms for estimation with pure DP in high dimensions

We've shown in the last couple of lectures that in high dimensions there are efficient algorithms for estimating the mean and covariance of a Gaussian in high dimensions under relaxed versions of differential privacy such as zCDP. While we've mostly focused on sample complexity during those lectures, one can also go back and check that all of these algorithms are also efficient, i.e. run in time which is polynomial in the dimension and the number of samples.

It turns out that the situation is more delicate when we instead ask for pure differential privacy. Consider the setting of mean estimation. The algorithm we sketched was the following: apply our 1D algorithm coordinate-wise, and then use advanced composition to argue that we don't lose too much in terms of privacy when we combine the outputs. To be more concrete, recall that to get ℓ_2 error α, we need to solve each univariate problem to accuracy $\alpha' = \alpha / \sqrt{d}$. Then, since we apply advanced composition to d subroutines, we need to set $\varepsilon' = \varepsilon / \sqrt{d}$ if we use to obtain (ε, δ)-DP. As discussed in the previous lecture this yields an overall sample complexity of

$$\widetilde{\Omega}\left(\frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon}\right) ,$$

where we have suppressed the dependence on δ for clarity.

However, if we insist upon pure differential privacy, we can no longer use advanced composition, and this yields a private sample complexity of

$$\widetilde{\Omega}\left(\frac{d}{\alpha^2} + \frac{d^{3/2}}{\alpha\varepsilon}\right) ,$$

in particular, the dependence on d is no longer linear. This sort of gap also seems to appear for covariance estimation, even in the well-conditioned setting. Here the main problem is that the Laplace mechanism adds entrywise noise which is proportional to the ℓ_1-sensitivity of the estimator, whereas the Gaussian mechanism adds entrywise noise which is proportional to the ℓ_2-sensitivity. However, the ℓ_1 sensitivity of the (truncated) empirical covariance is of order $\Theta(d^2)$, as opposed to $\Theta(d)$ for ℓ_2. This correspondingly causes a Frobenius loss of $\Theta(d^3)$ as opposed to $\Theta(d^2)$. In general, this is problematic, as generically ℓ_1-sensitivity will typically be larger by dimension-dependent factors than the ℓ_2-sensitivity.

It not trivial that this is avoidable, but recently, [1] gave pure DP algorithms for these problems that essentially match the runtime achieved for approximate differential privacy. However, these algorithms are not efficient—they run in time which is exponential in the dimension. A great open question is to resolve this gap:

Open Question 1.1. *Give a polynomial time $(\varepsilon, 0)$-DP algorithm which given samples from a Gaussian with unknown mean and known covariance (resp. known mean and unknown covariance), estimates the mean (resp. unknown covariance) at a rate that matches the rate for approximate DP.*

2 Local differential privacy

One potential drawback of differential privacy is that the users must still send their data to a central entity, and the privacy guarantee is satisfied only if this entity runs the private algorithm that they claim to run. In particular, if the data curator itself may be untrustworthy, then differential privacy may be insufficient. Local differential privacy [2, 3, 4], or LDP for short, is an attempt to rectify this problem. LDP is a stronger privacy guarantee, where, for any data point X, we release $\mathcal{A}(X)$, where $\mathcal{A}$ is a differentially private algorithm. The data curator then gets to see this privatized view of the data, and can output whatever it wants. By post-processing of differential privacy, the output of this algorithm will also preserve the privacy of the individual data points. Intuitively, we imagine that each user has local data (i.e. on their personal computer or smartphone), and then sends a privatized view of the data to the server, which may not be trustworthy. This way, the privacy-leaking data never leaves the user's local device.

Unsurprisingly, there is often a price that one must pay for LDP. As this is a stronger notion of privacy, the rates one is able to achieve are typically worse. For instance, consider the simple setting of mean estimation of a univariate Gaussian. Assume we have samples $X_1, \ldots, X_n \sim \mathcal{N}(\mu, 1)$, and let's even assume that $|\mu| \leq 1$, that is, we have a pretty decent estimate of μ. Then, to preserve $(\varepsilon, 0)$-LDP, we have to add noise to each X_i in a way which is $(\varepsilon, 0)$-differentially private. Since $|X_i| \lesssim \sqrt{\log n}$ with high probability, we can release $Y_i = X_i \cdot \mathbf{1}[|X_i| \lesssim \sqrt{\log n}] + Z_i$, where $Z_i \sim \mathrm{Lap}(\frac{\sqrt{\log n}}{\varepsilon})$, and the algorithm which takes X_i to Y_i will be $(\varepsilon, 0)$-differentially private (and this is essentially the best you can do). Given the Y_i, the best estimator is still essentially the empirical mean $\widetilde{\mu} = \frac{1}{n} \sum_{i=1}^n Y_i$. The cost of privacy is the magnitude of $\frac{1}{n} \sum_{i=1}^n Z_i$. Since this is a sum of n mean zero i.i.d. sub-exponential random variables with standard deviation $\approx \frac{\sqrt{\log n}}{\varepsilon}$, this will be of magnitude roughly $\approx \frac{\sqrt{\log n}}{\varepsilon \sqrt{n}}$, and so the rate of convergence of the overall algorithm is

$$|\mu - \widetilde{\mu}| = \Theta\left(\sqrt{\frac{1}{n}} + \frac{\sqrt{\log n}}{\varepsilon \sqrt{n}}\right) ,$$

with high probability, and moreover, one can show that this rate is optimal, up to logarithmic factors. In contrast, recall that for traditional $(\varepsilon, 0)$-differential privacy, we can achieve

$$|\mu - \widetilde{\mu}| = \Theta\left(\sqrt{\frac{1}{n}} + \frac{\sqrt{\log n}}{\varepsilon n}\right) ,$$

with high probability, for the same problem. The difference between these rates represents an additional price to pay for LDP as opposed to DP. Such a difference also manifests itself in high dimensional settings, see e.g. [5].

2.1 Shuffled differential privacy

One interesting attempt to bridge this gap, while still maintaining privacy without trusting the central party too much is an interesting notion called shuffled differential privacy [6]. In this paper, they show that much less noise needs to be added to each data point, if the central algorithm promises to perform a *permutation invariant* statistic, such as the kinds of statistics we've been considered like the empirical mean or the empirical covariance. In other words, if the central algorithm promises to perform a uniformly random shuffle of the data before performing its actual computation, then the individual users can get away with adding much less noise and still maintaining strong privacy guarantees. Of course, this still requires some degree of trust on the part of the users, but perhaps less so. Moreover, it is not implausible that the fact that such a permutation has been applied could be verified by some cryptographic protocol.

3 Private stochastic optimization

We have so far focused primarily on unsupervised learning tasks. This raises the following natural question: what about supervised learning? As in the case for robust statistics, we can study this largely through the lens of stochastic optimization. Recall that stochastic optimization is the following problem: we have a distribution D over functions $f(w)$, and given samples from this distribution (which is our data), the goal is to minimize $R(w) = \mathbb{E}_{w \sim D}[f(w)]$, i.e. the expected loss of our model w. As it turns out, in many settings, noisy SGD gives optimal rates. Formally, consider the following algorithm: suppose we have samples $\{f_1, \ldots, f_n\}$, and suppose that $w \in \mathcal{W}$ for some convex set $\mathcal{W} \subseteq \mathbb{R}^d$. Choose a starting point $w_0 \in \mathcal{W}$ arbitrarily. Then, for $t = 1, \ldots, T$ iterations:

- Sample a batch B_t by sampling each point with probability m/n with replacement.
- Let
$$w_{t+1} = \Pi_{\mathcal{W}} \left(w_t - \eta \cdot \left(\frac{1}{m} \sum_{j \in B_t} \nabla f_j(w_t) \cdot \mathbf{1}\left[\|\nabla f_j(w_t)\|_2 \leq L \right] + g_t \right) \right) ,$$
where $\Pi_{\mathcal{W}}$ is projection onto $\mathcal{W}$, η is a step-size parameter, L is a bound on the Lipschitz constant of the functions, and $g_t \sim \mathcal{N}(0, \sigma^2 I)$ are independent, where $\sigma > 0$ is the privacy parameter.

To be precise, let us call this algorithm $\mathrm{SGD}(m, \eta, \sigma, T, L)$. Then, we have the following theorems:

Theorem 3.1 ([7])**.** *Let $\varepsilon \in (0,1)$, $\delta \leq 1/n^2$, and $T \geq 1$. Let f be L-Lipschitz for all $f \in \mathrm{supp}(D)$, and let $\sigma \gtrsim \frac{L\sqrt{T \log 1/\delta}}{n\varepsilon}$. Then, for any m, η, we have that $\mathrm{SGD}(m, \eta, \sigma, T, L)$ is (ε, δ)-DP.*

Moreover, this noisy SGD algorithm is still guaranteed to converge, as the gradient estimates are still unbiased estimates of the true gradient (as we are simply adding zero-mean gaussian noise). In [8], the authors work out precise and optimal rates by combining Theorem 3.1 with techniques from stochastic optimization. To prove Theorem 3.1, [7] develop a new method for accounting for privacy loss they call *moments accountant method*. However, we can get a slightly sloppier bound by essentially only using techniques we already know:

Theorem 3.2. *Let $\varepsilon \in (0,1)$, $\delta \leq 1/n^2$, and $T \geq 1$. Let f be L-Lipschitz for all $f \in \mathrm{supp}(D)$, and let $\sigma \gtrsim \frac{L\sqrt{T \log 1/\delta \log T/\delta}}{n\varepsilon}$. Then, for any m, η, we have that $\mathrm{SGD}(m, \eta, \sigma, T)$ is (ε, δ)-DP.*

We do need one, relatively straightforward tool, namely, that subsampling improves privacy:

Lemma 3.3 ([4])**.** *Let S be a dataset, and let T be a dataset formed by sampling each element of S independently with probability p. Suppose $\mathcal{A}$ is an (ε, δ)-DP algorithm when given T. Then the algorithm $\mathcal{A}'$ which (1) first subsamples T from S, then (2) outputs $\mathcal{A}(T)$ is $(q\varepsilon, q\delta)$-DP for S.*

Proof of Theorem 3.2. Each mini-batch gradient has ℓ_2 sensitivity at most L/m. As a result, by our choice of parameters, each individual gradient plus Gaussian noise is $\left(\frac{n\varepsilon}{m\sqrt{T \log 1/\delta}}, \delta/T \right)$-DP for B_t. By combining this with Lemma 3.3 and advanced composition, we obtain that the overall algorithm is (ε, δ)-DP. □

3.1 Better iterative guarantees for DP

An unattractive phenomena of these bounds is that our privacy loss scales poorly with the number of iterations we take, which is a problem for many real-world algorithms. However, one could hope that this is not really the case. Intuitively, one shouldn't pay very much (or at all) in terms of privacy for many iterations of iterative methods, as all the information gets "smushed" together, and so iterating an algorithm should have the effect of not compounding privacy loss, but perhaps amplifying it. This is unfortunately not true in general, as far as we can tell, however, in the case of contractive algorithms, [9] demonstrates that this is true for a slightly non-standard notion of privacy. A great open question is to characterize more generally when this sort of phenomena occurs.

References

[1] Mark Bun, Gautam Kamath, Thomas Steinke, and Zhiwei Steven Wu. Private hypothesis selection. *arXiv preprint arXiv:1905.13229*, 2019.

[2] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[3] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222. ACM, 2003.

[4] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[5] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[6] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

[7] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[8] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. In *Advances in Neural Information Processing Systems*, pages 11279–11288, 2019.

[9] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 521–532. IEEE, 2018.