CS839: Differential Privacy and Learning March 24, 2026

19: Stable Covariance

Instructor: Gavin Brown *Scribe:* Gavin Brown

Disclaimer: This document is intended as an informal supplement to in-class note-taking. It has not been given the level of scrutiny expected in polished lecture notes, let alone that reserved for peer-reviewed publications.

Today we will see (parts of) a technique for differentially private covariance estimation. Last class we saw the FriendlyCore method as applied to estimating the mean of a Gaussian distribution. It had four main steps:

- Preprocess the data, removing outliers via soft weights in $[0, 1]$.
- Privately check that not too many outliers were removed.
- Produce a weighted estimate (which is not private, but is stable).
- Add noise.

The approach we'll see today follows these same steps, although they will be more complex. We will not get close to proving everything, but an informal version of the main result is as follows.

Theorem 0.1 (Informal)**.** *There is an efficient (ε, δ)-differentially private algorithm that, given n i.i.d. samples from a distribution $\mathcal{N}(0, \Sigma)$ with $\Sigma \succ 0$, with high constant probability returns a matrix $\tilde{\Sigma}$ such that*

$$\|\Sigma^{-1/2}(\tilde{\Sigma} - \Sigma)\Sigma^{-1/2}\|_2 \leq \tilde{O}\left(\sqrt{\frac{d}{n}} + \frac{d^{3/2}\sqrt{\log(1/\delta)}}{\varepsilon n}\right)$$

as long as $n \gtrsim d\log(1/\delta)/\varepsilon$.

Note that this theorem directly gives guarantees in Frobenius norm using the $\|\cdot\|_F \leq \sqrt{d}\|\cdot\|_2$ inequality.

1 Warm-Up: Well-Conditioned Gaussians

[This section was not covered in class.]

The sample complexity guarantees of Theorem 0.1 match what we would get if we had strong a priori bounds on the range of eigenvalues of Σ. In this setting, a simple clip-and-noise algorithm is close to optimal [Dwork et al., 2014, Kamath et al., 2019].

Algorithm 1 Well-Conditioned Covariance Estimator

Input: data $x_1, \ldots, x_n \in \mathbb{R}^d$; clipping threshold $c > 0$; noise scale $\sigma > 0$
Returns: $\tilde{\Sigma} \in \mathbb{R}^{d \times d}$

1: $S \leftarrow \{i \in [n] : \|x_i\|_2^2 \leq c\}$
2: Sample $N \sim \mathrm{GUE}(\sigma^2)$
3: Return $\tilde{\Sigma} \leftarrow N + \frac{1}{n}\sum_{i \in S} x_i x_i^T$

Here GUE stands for *Gaussian unitary ensemble*: $N \sim \mathrm{GUE}(\sigma^2)$ means $N \in \mathbb{R}^{d\times d}$ has entries $N_{ij} \sim \mathcal{N}(0, \sigma^2)$ independently if $i \le j$, and $N_{ij} = N_{ji}$ otherwise (that is, N is symmetric with its upper triangle being i.i.d. Gaussian).

Kamath et al. [2019] prove (formal) versions of the following claims. It's a good exercise to work through the proofs on your own, but ultimately they require concentration inequalities that you'll presumably need to look up.

Claim 1.1 (Informal)**.** *If $\sigma \gtrsim \frac{c\sqrt{\log(1/\delta)}}{n\varepsilon}$, then Algorithm 1 is (ε, δ)-DP.*

Claim 1.2 (Informal)**.** *Suppose $x_1, \dots, x_n \sim \mathcal{N}(0, \Sigma)$ for $\mathbb{I} \preceq \Sigma \preceq 2\mathbb{I}$. Then Algorithm 1 with $c \approx d$ and $\sigma \approx \frac{c\sqrt{\log(1/\delta)}}{n\varepsilon}$ with high constant probability returns $\tilde{\Sigma}$ such that*

$$\|\Sigma^{-1/2}(\tilde{\Sigma} - \Sigma)\Sigma^{-1/2}\|_2 \le \tilde{O}\left(\sqrt{\frac{d}{n}} + \frac{d^{3/2}\sqrt{\log(1/\delta)}}{\varepsilon n}\right).$$

2 Gaussian Sampling Mechanism

We start with the last step and discuss the noise mechanism. Let's suppose we have a non-private estimate $\hat{\Sigma}(X)$ of the covariance of our dataset $X \in \mathbb{R}^{n\times d}$. How can we privatize it?

A first attempt would be to introduce additive noise, e.g., add appropriately scaled Gaussian noise to each entry in the matrix. Under certain assumptions this will yield good results, but the fundamental drawback is that the noise does not respect the shape of the matrix. Informally, we will add a lot of noise in the "small" directions of the matrix. There are lots of more-sophisticated additive noise mechanisms, but they all have the same basic issue. One can take an additive noise mechanism and iteratively refine the estimate it gives [Kamath et al., 2019], but this won't quite get the guarantees we want.

Instead, we will not use additive noise. We will use the *Gaussian sampling mechanism* of Alabi et al. [2023], Algorithm 2. It returns an estimate which matches the geometry of the non-private estimate. The main parameter is m, the number of "synthetic" samples drawn. As $m \to \infty$ we have a better estimate (since $\tilde{\Sigma} \approx \hat{\Sigma}(X)$) but also lose privacy. So we will try to set m as large as possible.

Algorithm 2 Gaussian Sampling Mechanism

Input: $\hat{\Sigma} \in \mathbb{R}^{d\times d}$, $m \in \mathbb{N}$
Returns: $\tilde{\Sigma} \in \mathbb{R}^{d\times d}$
1: Sample $y_1, \dots, y_m$ i.i.d. from $\mathcal{N}(0, \hat{\Sigma})$
2: Return $\tilde{\Sigma} = \frac{1}{m}\sum_{i=1}^m y_i y_i^T$

To state the guarantees, we will use the following notion of distance between two matrices.

$$d_M(\Sigma_1, \Sigma_2) := \max\left\{\|\Sigma_2^{-1/2}(\Sigma_1 - \Sigma_2)\Sigma_2^{-1/2}\|_F, \|\Sigma_1^{-1/2}(\Sigma_1 - \Sigma_2)\Sigma_1^{-1/2}\|_F\right\}.$$

The "M" stands for *Mahalanobis* and the matrix norm is Frobenius. If either of the matrices is not invertible we will call the distance $+\infty$. To interpret it, if $d_M(\Sigma_1, \Sigma_2)$ is small then $\Sigma_2^{-1/2}\Sigma_1\Sigma_2^{-1/2} \approx \mathbb{I}$.

We have the following guarantee for DP.

Claim 2.1. *Fix $\varepsilon, \delta \in (0,1)$ and positive definite Σ_1, Σ_2. If Σ_1, Σ_2 satisfy*

$$d_M(\Sigma_1, \Sigma_2) \le \Delta \le \frac{\varepsilon}{8\log(1/\delta)},$$

then for any $m \le \frac{\varepsilon^2}{8\Delta^2 \log(1/\delta)}$, Algorithm 2's output satisfies $A(\Sigma_1, m) \approx_{\varepsilon,\delta} A(\Sigma_2, m)$.

The rest of this lecture will drive toward building a non-private covariance estimator $\hat{\Sigma}$ which is stable in the sense that, on adjacent X, X', it returns $\hat{\Sigma}(X)$ and $\hat{\Sigma}(X')$ such that

$$d_M(\hat{\Sigma}(X), \hat{\Sigma}(X')) \le O(d/n).$$

(Formally, the estimator will either satisfy that inequality or return FAIL, just like how in FriendlyCore the algorithm either returns a weighted mean or fails.)

3 Leverage Filtering

Our task now is to construct an estimator $\hat{\Sigma}(X)$ of the covariance which is stable, i.e., for any adjacent datasets x, x' we can bound $d_M(\hat{\Sigma}(X), \hat{\Sigma}(X'))$. As with FriendlyCore, we will do this by finding removing outliers. We use the following quantity to determine what is an outlier.

Definition 3.1 (Leverage Score)**.** For dataset $X \in \mathbb{R}^{n \times d}$ and index $i \in [n]$, the *i-th leverage score* is

$$\ell(i) = x_i^T (X^T X)^{-1} x_i.$$

For a subset $S \subseteq [n]$ we use $\ell_S(i) = x_i^T (X_S^T X_S)^{-1} x_i$ to denote the leverage on a subset, where $X_S \in \mathbb{R}^{|S| \times d}$ denotes X with the rows not in S removed.

If $X^T X$ is not invertible we will say $\ell(i) = \infty$, although a lot of what follows can be generalized with the pseudoinverse. Leverage is a classical notion in statistics and also plays a fundamental role in numerical linear algebra. It has a lot of interpretations, but for now we will point out that it is like a rescaled Euclidean norm: if $\bar{\Sigma} = \frac{1}{n} X^T X$ is the empirical covariance[1] then

$$\begin{aligned} \ell(i) &= x_i^T (X^T X)^{-1} x_i \\ &= \frac{1}{n} \cdot x_i^T \bar{\Sigma}^{-1} x_i \\ &= \frac{1}{n} \cdot \|\bar{\Sigma}^{-1/2} x_i\|_2^2. \end{aligned}$$

We will filter out points with large leverage score, which corresponds to throwing out points that have large ℓ_2 norms after whitening by the empirical covariance. If the data are i.i.d. from a Gaussian distribution $\mathcal{N}(0, \mathbb{I})$ then once n is large enough we expect $\bar{\Sigma} \approx \mathbb{I}$ and thus would expect $\ell(i) \approx \frac{1}{n} \cdot \|x_i\|_2^2 \approx \frac{d}{n}$.

If I asked you to come up with an algorithm for filtering out high-leverage points, you would probably write down Algorithm 3. It removes high-leverage points, then recomputes the remaining leverage scores and repeats.

[1] We are going to ignore the step of centering the data.

Algorithm 3 Greedy Leverage Filter

Input: Data $X \in \mathbb{R}^{n \times d}$, leverage threshold $L > 0$
Returns: $S \subseteq [n]$
1: $\texttt{OUT} \leftarrow \{i \in S \mid \ell_S(i) > L\}$
2: $S \leftarrow S \setminus \texttt{OUT}$
3: Return S

This algorithm itself will not work: the greedy process is inherently unstable. One can construct[2] pairs of datasets X, X' and thresholds L where nothing is removed under X and many points are removed under X'.

We will need one nice fact about this algorithm. Its proof is left as an exercise to the reader.

Lemma 3.2. *Fix X, L and suppose S satisfies: for all $i \in S$, $\ell_S(i) \le L$. Let T be the output of Algorithm 3. Then $S \subseteq T$.*

As an immediate consequence, we see that there is always a unique largest "L-good subset" and Algorithm 3 finds it.

4 Stabilizing the Greedy Filter

We now provide a stabilizing wrapper around the greedy algorithm.

Algorithm 4 Stable Leverage Filter

Input: Data $X \in \mathbb{R}^{n \times d}$, leverage threshold $L_0 > 0$, parameter $k \in \mathbb{N}$
Returns: $\hat{d} \in \mathbb{N}$, $\vec{w} \in [0,1]^n$
1: **for** $j = 0, 1, \ldots, 2k$ **do**
2: $\quad S_j \leftarrow \texttt{GreedyLeverageFilter}(X, e^{j/k} L_0)$
3: **end for**
4: $\hat{d} \leftarrow \min\{k, \min_{0 \le j \le k}\{n - |S_j| - j\}\}$
5: $\forall i \in [n]$, $w_i \leftarrow \frac{1}{k} \sum_{j=k+1}^{2k} 1\{i \in S_j\}$
6: Return $\hat{d}, \vec{w}$

This is slightly mysterious. We find $2k+1$ largest "good subsets" at a number of increasing leverage thresholds. We use the subsets $0, 1, \ldots, k$ to compute some integer $\hat{d}$ and the subsets $k+1, \ldots, 2k$ to compute some weights $\vec{w}$. Before stating the actual guarantees, a few remarks to orient ourselves.

- $\hat{d}$ will serve as an approximation of the number of outliers removed, analogous to $\|\vec{x}\|_1$ in FriendlyCore. Smaller $\hat{d}$ means the dataset is better.
- The vector $\vec{w}$ will be our weights, as in FriendlyCore. Our stable estimate of the covariance will be $\frac{1}{n} \sum_i w_i \cdot x_i x_i^T$.
- If there is a single point x_i that has very high leverage, then it will always be removed by the greedy algorithm and will receive $w_i = 0$ weight.
- If L_0 is very large, then every call to the greedy algorithm will return $S_j = [n]$. We will compute $\hat{d} \le \min_{0 \le j \le k}\{n - |S_j| - j\} \le n - |S_0| - 0 = 0$ and $\vec{w} = \vec{1}$.

[2] Try it yourself!

- If L_0 is very small, then every call to the greedy algorithm will return $S_j = \emptyset$. We will compute $\hat{d} = k$ and $\vec{w} = \vec{0}$.

Ultimately, we will need the following claims.

1. **Score Stability:** $\forall X \sim X'$, $|\hat{d}(X) - \hat{d}(X')| \leq 2$.
2. **Completeness:** if $\ell(i) \leq L_0$ for all i, then $\hat{d}(X) = 0$ and $\vec{w}(X) = \vec{1}$.
3. **Soundness:** if $w_i > 0$, then $x_i^T \left(\frac{1}{n}\sum_i w_i \cdot x_i x_i^T\right)^{-1} x_i \leq 20L_0$.
4. **Weight Stability:** for all $X \sim X'$, if $\hat{d}(X), \hat{d}(X') < k$, then $\|w(X) - w(X')\|_1 \leq 2$.

We will not prove any of these, but we will prove the key lemma for Algorithm 4, which allows us to reason about how it behaves on adjacent datasets.

Lemma 4.1. *Fix $X \in \mathbb{R}^{n\times d}$ and $S \subseteq [n]$. Suppose for all $i \in S$ we have $\ell_S(i) \leq L$. Then for all $i, j \in S$,*

$$\ell_{S\setminus\{j\}}(i) \leq e^L \ell_S(i).$$

Since we expect $L \approx d/n$ and are in a regime where $n \gtrsim d$, this lemma means that removing a single point from a "good" subset can only lead to a modest multiplicative increase in any other leverage score.

Proof. We use the ever-helpful Sherman–Morrison formula for rank-one updates to a matrix inverse:

$$(A - vv^T)^{-1} = A^{-1} + \frac{A^{-1}vv^TA^{-1}}{1 + v^TA^{-1}v}.$$

Let $A = \sum_{j\in S} x_j x_j^T$, so by assumption we have $\ell_S(i) = \|A^{-1/2}x_i\|_2^2 \leq L$.

We bound the leverage after removal.

$$\begin{aligned}
\ell_{S\setminus\{j\}}(i) &= x_i^T \left(\sum_{j\in S}(x_k x_k^T) - x_j x_j^T\right)^{-1} x_i \\
&= x_i^T \left(A - x_j x_j^T\right)^{-1} x_i \\
&= x_i^T A^{-1} x_i + \frac{x_i^T A^{-1} x_j \cdot x_j^T A^{-1} x_i}{1 + x_j^T A^{-1} x_j}.
\end{aligned}$$

Cauchy–Schwarz allows us to bound the numerator:

$$|x_i^T A^{-1} x_j|^2 = |\langle A^{-1/2}x_i, A^{-1/2}x_j\rangle|^2 \leq \|A^{-1/2}x_i\|_2^2 \|A^{-1/2}x_j\|_2^2 \leq \ell_S(i)\cdot L.$$

The denominator is at least one. So we have $\ell_{S\setminus\{j\}}(i) \leq \ell_S(i) + L\cdot\ell_S(i) \leq e^L\ell_S(i)$. □

5 Further Reading

These notes present the work of Brown et al. [2023] in a way designed to parallel the techniques of Tsfadia et al. [2022]. Concurrent work of Duchi et al. [2023] gave similar algorithms; they did not analyze them for covariance estimation, but they would work similarly.

Later in class we will talk about lower bounds for statistical estimation. The technique of *fingerprinting lower bounds* leads to allows one to show that these results achieve the optimal relationships between d, n, and ε. (As with FriendlyCore, the main theorem includes a $\sqrt{\log(1/\delta)}$ term that is not known to be necessary, although optimality is not fully settled.)

Beyond DP covariance, the techniques in this lecture are used in recent work on mean estimation [Duchi et al., 2023, Brown et al., 2023], least squares [Brown et al., 2024], and private sampling from distributions [Iverson et al., 2025]. The stable covariance estimator discussed here is useful as a first subroutine. Also, the blueprint of "greedy filter outlier + stabilize" is applied in a similar form. It will likely continue to be useful.

References

Daniel Alabi, Pravesh K Kothari, Pranay Tankala, Prayaag Venkat, and Fred Zhang. Privately estimating a gaussian: Efficient, robust and optimal. In *Proceedings of the fifty-fifth annual ACM symposium on Theory of computing*, 2023.

Gavin Brown, Samuel Hopkins, and Adam Smith. Fast, sample-efficient, affine-invariant private mean and covariance estimation for subgaussian distributions. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5578–5579. PMLR, 2023.

Gavin Brown, Jonathan Hayase, Samuel Hopkins, Weihao Kong, Xiyang Liu, Sewoong Oh, Juan C Perdomo, and Adam Smith. Insufficient statistics perturbation: Stable estimators for private least squares extended abstract. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 750–751. PMLR, 2024.

John Duchi, Saminul Haque, and Rohith Kuditipudi. A pretty fast algorithm for adaptive private mean estimation. In *Conference on Learning Theory*, pages 2511–2551. Proceedings of Machine Learning Research, 2023.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 11–20, 2014.

Valentio Iverson, Gautam Kamath, and Argyris Mouzakis. Optimal differentially private sampling of unbounded gaussians. In *The Thirty Eighth Annual Conference on Learning Theory*, pages 2893–2941. PMLR, 2025.

Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. In *Conference on Learning Theory*, pages 1853–1902. PMLR, 2019.

Eliad Tsfadia, Edith Cohen, Haim Kaplan, Yishay Mansour, and Uri Stemmer. Friendlycore: Practical differentially private aggregation. In *International Conference on Machine Learning*, pages 21828–21863. PMLR, 2022.