36-709: Advanced Statistical Theory I **Spring 2020**

Lecture 6: January 30

Lecturer: Siva Balakrishnan *Scribe: Don Dennis*

6.1 Matrix Estimation

Over the next few lectures we will be talking about the problem of matrix estimation. Today, we will start by introducing multiple setups under which one would want to estimate a matrix and elaborate on one of these.

6.1.1 Covariance Estimation

We talked a little bit about covariance estimation in lecture 1; given random matrices $X_1, X_2, \ldots, X_n \overset{\text{iid}}{\sim} P$, our objective is to estimate the covariance matrix,

$$\Sigma = \mathbb{E}[XX^T].$$

The natural estimator for this problem is the sample covariance,

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} X_i X_i^T.$$

This is an average of 'simple', rank 1 matrices. We will often use this setup as a guiding example when thinking about averages of random matrices. In a future lecture, we will also analyse how close averages of random matrices are to their expectation.

6.1.2 Matrix Sensing

In this setup, we observe the matrix X_i and a response y_i) of the form

$$y_i = \text{tr}({X_i}^T M^*) + \epsilon_i,$$

and our objective is to estimate the matrix M^*. You should think of this as the analogue of the regular vector regression problem,

$$y_i = \beta^T x_i + \epsilon_i,$$

where our objective is estimating the vector β. In fact, if we rewrite the matrix sensing problem in terms of vectorized forms of the matrices M^* and X_i, we get the regular vector regression model.

$$\begin{aligned} y_i &= \text{tr}({X_i}^T M^*) + \epsilon_i \\ &= \text{vec}(X_i)^T \text{vec}(M^*) + \epsilon. \end{aligned}$$

The difference between the two problems is in the type of assumptions we make. For matrix sensing we will impose some (spectral) structure on the matrix M^* and will exploit that.

The canonical example of the matrix sensing is the matrix completion problem. In matrix completion, we observe a subset of the entries of the matrix M^* (potentially with some noise) and are tasked to estimating M^*. Thus, each X_i will be of the form,

$$\Big(X_i\Big)_{(m,n)} = \begin{cases} 0 & (m,n) \neq (j,k) \\ 1 & (m,n) = (j,k) \end{cases},$$

such that $y_i = M^*_{j,k} + \epsilon_i$.

6.1.3 Analogue of Gaussian Sequence Model (GSM)

You should think about this as the analogue of the Gaussian Sequence Model. Recall that in the GSM model, we observe a noisy vector y and want to denoise it. That is, we observe y of the form

$$y = \theta^* + \epsilon,$$

for some zero mean Gaussian ϵ and our objective is recovering θ^*.

In the matrix version of this problem, we observe some matrix Y which has the form,

$$Y = M^* + W,$$

where M^* is the matrix we want to estimate and W is a matrix with zero mean, sub-gaussian entries. We will try to argue how well we can estimate M^* under various assumptions.

6.2 Analysing the Gaussian Sequence Model

Before we analyse the GSM mode, we will look at a few special cases of it.

6.2.1 Special Cases

6.2.1.1 Probability Matrix

Let us consider the special case where each entry of M^* is a probability and the corresponding entries of Y are rounded version of these probabilities. That is,

$$Y_{i,j} \in \{0, 1\} \qquad \text{and} \qquad M^*_{i,j} = P(Y_{i,j} = 1).$$

You can imagine determining the value of each entry of Y by tossing a biased coin whose bias is determined by the corresponding entry in M^*. Thus each entry $Y_{i,j} \sim \text{Ber}(M^*_{i,j})$.

We claim that this is in the matrix GSM form

$$Y = M^* + W,$$

if we let

$$W_{i,j} = \begin{cases} -M^*_{i,j} & w.p. \quad 1 - M^*_{i,j} \\ 1 - M^*_{i,j} & w.p. \quad M^*_{i,j} \end{cases}.$$

For our claim to hold, we need to show that W has mean zero sub-gaussian entries.

Since $M^*_{i,j}$ is bounded, it is also sub-gaussian. As for the mean, observe that

$$\mathbb{E}[W_{ij}] = -M^*_{i,j}(1 - M^*_{i,j}) + (1 - M^*_{i,j})(M^*_{i,j}) = 0,$$

thus proving our claim. Lets now see some examples where this particular version of the matrix GSM is studied.

Example: Community Detection (Stochastic block model)

Say we have n objects that form a graph between them. There is a notion of "communities" in this graph, in the sense that two nodes belonging to the same community are more likely to have an edge between them than two nodes belonging to different communities. As a motivating example, consider the nodes as being people and edges representing friendships. In this setup, two individuals who belong to the same community are more likely to form a friendship compared to two individuals from different communities.

We can model this graph as an $n \times n$ matrix M where $M_{i,j}$ represents the probability that there is an edge between node i and j in the graph. Consider nodes i, j and k from the community graph. Let node j belong to the same community as i and node k belong to a different community as i. Then,

$$M_{i,j} > M_{i,k}.$$

In the community detection problem, we will observe some noisy version of the matrix M and our objective is to recover the communities.

Example: Graphon Estimation

Consider a smooth function that maps the unit square to a probability. That is,

$$f : [0, 1]^2 \to [0, 1].$$

Lets evaluate f on an $n \times n$ grid and populate an $n \times n$ matrix M^* such a that M_{ij} takes the value of f evaluated at the ij-th grid point.

We will revisit graphon estimation later but for now, we can consider the it as modelling the estimation of a smooth matrix. A natural way to say a matrix is smooth in some sense is to assert that the entires arise from a smooth function. Here, we will observe a permuted and noisy version of M^* and our objective will be to recover the true M^*.

Example: Estimating Strongly Stochastically Transitive Matrix (SST)

SST matrices arise in ranking problems or, more broadly, in problems that involve pair wise comparisons. The matrix M^* captures some notion of the result of this pairwise comparison or preference. For example, consider the case of sports teams indexed by i. In this case, M^*i, j could represent the probability that the team i defeats a team j in a match. Another example is from ranking where $M^*_{i,j}$ could be interpreted as the probability that some object i is preferred over object j. Lets represent the notion of preference of i over j as $P(i > j)$. Then, if $M^*_{i,j} = P(i > j)$, then $M^*_{j,i} = 1 - M^*i, j$. Further $M^*_{i,i} = 0.5$.

Our objective as before is estimating the matrix M^* from a noise, possibly perturbed observation. In the general case this is a difficult to do and we will often model the problem under some structural assumptions. One classic often used model is the Bradley Terry Luce (BTL) model. Under the BTL model, we associate a 'quality' β_i with each object i which determines how preferable they are. The BTL model then says that

$$\begin{aligned} P(i > j) &\propto \exp(\beta_i - \beta_j) \\ &= \frac{\exp(\beta_i - \beta_j)}{\exp(\beta_i - \beta_j) + \exp(\beta_j - \beta_i)}. \end{aligned}$$

Thus $P(i > j) \to 1$ if $\beta_i \gg \beta_j$.

The BTL model impose constrains on pair wise probabilities. The more general modeling framework assumes some unknown ranking of objects Π^* that defines preference. That is, for two objects i and j if $\Pi^*(i) \geq \Pi^*(j)$ then i is preferred over j. Further, the if $\Pi^*(i) > \Pi^*(j)$, then probability that i is better than any other k should be greater than the probability that j is better than k thus introducing a notion of transitivity. This particular model induces a particular type of matrix M^* called a strongly stochastically transitive matrix.

All of these are examples of probability matrices, that we observe in noise, and wish to estimate. In the next section we will present a simple estimator for this task, and analyze it.

6.2.2 Analysis

Let us return to the general matrix GSM model. We want to estimate a matrix M^* from a noisy observation Y,

$$Y = M^* + W.$$

Recall that entries of the matrix W are zero mean sub-gaussians. We will propose an estimator and analyse its performance.

Assume Y, M^*, W are $n \times n$ and symmetric. Define our estimator $\widehat{M}$ as

$$\widehat{M} = \sum_{i=1}^{n} \beta_i I\{|\beta_i| \geq 2t\} u_i {u_i}^T$$

where $Y = \sum \beta_i u_i {u_i}^T$ is the spectral decomposition of Y. That is, we are using $\widehat{M}$, a low rank approximation of Y as an estimator for M^*. The intuition for this choice comes form the fact that estimating the singular vectors and values of M^* that are comparable to $\|W\|_{op}$ will be difficult because of low 'signal to noise ratio'.

We claim that if $t = C\sqrt{n + \log(1/\delta)}$, then with probability at least $1 - \delta$,

$$\|\widehat{M} - M^*\|_F^2 \leq C \sum_{i=1}^{n} \min\{t^2, \alpha_i^2\}$$

where $M^* = \sum_{i=1}^{n} \alpha_i v_i {v_i}^T$ is the spectral decomposition of M^*. Additionally if M^* is rank r, we also have

$$\|\widehat{M} - M^*\|_F^2 \leq Cr\{n + \log(1/\delta)\}.$$

Note that our estimator is not aware of the rank of M^*.

Lets proceed to proving these claims. We will only prove a part of this claim in this lecture and will assume that for our choice of t, with probability $1 - \delta$, $\|W\|_{op} \leq t$.

We also require Weyl's matrix perturbation inequality for the proof: let the spectral decompositions for M^* and Y be,

$$M^* = \sum_{i=1}^{n} \alpha_i v_i {v_i}^T \qquad \alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n$$

$$Y = \sum_{i=1}^{n} \beta_i u_i {u_i}^T \qquad \beta_1 \leq \beta_2 \leq \ldots \leq \beta_n$$

then Weyl's inequality states that

$$|\alpha_i - \beta_i| \leq \|W\|_{op} \qquad \forall i.$$

We now proceed with the proof. Let $S = \{i : |\beta_i| \geq 2t\}$ and let M_S^* represent the corresponding approximation of M^*,

$$M_S^* = \sum_{i \in S} \alpha_i v_i {v_i}^T.$$

Then,

$$\|\widehat{M} - M^*\|_F \leq \|\widehat{M} - M_S^*\|_F + \|M_S^* - M^*\|_F.$$

Lets consider each of these terms separately.

$$\|M_S^* - M^*\|_F = \|M_S^* - (M_S^* + M_{S^c}^*)\|_F = \|M_{S^c}^*\|_F = \sqrt{\sum_{i \in S^c} \alpha_i^2},$$

and,

$$\begin{aligned}\|\widehat{M} - M_S^*\|_F &\leq \sqrt{\text{rank}(\widehat{M} - M^*)}\|\widehat{M} - M_S^*\|_{op} \\ &\leq \sqrt{2|S|}\Big(\underbrace{\|\widehat{M} - Y\|_{op}}_{\text{(a)}} + \underbrace{\|Y - M^*\|_{op}}_{\text{(b)}} + \underbrace{\|M^* - M_S^*\|_{op}}_{\text{(c)}}\Big).\end{aligned}$$

Again we bound these terms individually. For (a),

$$\begin{aligned}\widehat{M} &= \sum_{i=1}^{n} \beta_i I\{|\beta_i| \geq 2t\} u_i {u_i}^T \\ Y &= \sum \beta_i u_i {u_i}^T.\end{aligned}$$

Therefore,

$$\|\widehat{M} - Y\|_{op} \leq 2t.$$

From our assumption on the noise, we can upper bound (b) as,

$$\|Y - M^*\|_{op} = \|W\|_{op} \leq t.$$

And finally for (c),

$$\|M^* - M_S^*\|_{op} = \|M_{S^c}^*\|_{op} = \max_{i \in S^c} |\alpha_i|.$$

We establish an upper bound for the above by contradiction. Assume that for some $i \in S^c$, $\alpha_i \geq 3t$. Then Weyl's inequality combined with our assumption that $\|W\|_{op} \leq t$ would require $\beta_i \geq 2t$. But this would imply $i \in S$ leading to a contradiction. Thus for all $i \in S^c$, $|\alpha_i| < 3t$ and hence

$$\|M^* - M_S^*\|_{op} \leq 3t.$$

Putting all this together (ignoring constants), we get

$$\begin{aligned}\|\widehat{M} - M^*\|_F &\leq C\left[\sqrt{|S|}t + \sqrt{\sum_{i\in S^c}\alpha_i^2}\right] \\ &\leq 2C\left[\sqrt{|S|t^2 + \sum_{i\in S^c}\alpha_i^2}\right] \\ &\leq 2C\left[\sqrt{\sum_{i=1}^n \min\{t^2, \alpha_i^2\}}\right].\end{aligned}$$

Here, the second to last inequality used the fact that $\sqrt{a} + \sqrt{b} \leq 2\sqrt{a+b}$.

Oracle inequality: Notice that we can rewrite the above error bound to reveal a decomposition between the approximation error and an estimation error. Concretely, notice that we have shown,

$$\begin{aligned}\|\widehat{M} - M^*\|_F^2 &\lesssim \sum_{i=1}^n \min\{t^2, \alpha_i^2\} \\ &\lesssim \|M^* - M_S^*\|_F^2 + |S|n,\end{aligned}$$

where M_S^* is the best rank $|S|$ approximation to M^*. Equivalently we may write this as:

$$\|\widehat{M} - M^*\|_F^2 \leq C \inf_{\Theta}\{\underbrace{\|M^* - \Theta\|_F^2}_{\text{approx error}} + \underbrace{\text{rank}(\Theta)n}_{\text{estimation error}}\}.$$

This is referred to as the oracle inequality because it shows that our estimator trades-off optimally the estimation error and approximation error (in essence does as well as trying to estimate the best rank-r approximation to M^* without needing to know r a-priori).