CSE599i: Online and Adaptive Machine Learning **Winter 2018**

Lecture 7: Regression

Lecturer: Kevin Jamieson *Scribes: Omid Sadeghi, Johannes Linder, Felix Leeb, Sumit Mukherjee*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

This script reviews optimal design in linear experiments, where the goal is to pick a small number of *design points* (input data points, or feature vectors) from the space of possible data points so that the underlying model, which relates input variables to output responses, is estimated accurately. The chosen set of input data points makes up one particular *design*, and the problem of finding an optimal (small) design is of interest in many regression settings. For example, in an experiment there might be substantial cost associated with measuring the response of each trial (input vector). The experimenter could then employ optimal design to estimate the model using as few trials as possible. Another popular use case is in the big data domain, where it might be computationally intractable to perform regression on the entire data set. Optimal design can be used to select a subset of the data that estimates the model efficiently. Also, as is reviewed in the scribe, optimal design has applications in bandit problems, especially linear bandits. This script aims to serve as scribe notes for the seventh lecture of CSE 599, with extensions mainly based on the work of [1].

1 Linear Experimental Design

The section starts by defining Linear Regression, which in Linear experimental design we assume is an accurate model for describing the relationship between the input features and corresponding output values. After that, the Experimental design problem is stated and related to various criteria one typically evaluates a chosen design by. Finally, we review the work of [1] on randomized sampling-based algorithms for achieving provably bounded, and in some aspects optimal, designs.

1.1 Linear Model

Consider the problem of estimating a vector of unknown parameters $\beta \in \mathbb{R}^p$ from observed measurements or experiments $\{x_i, y_i\}_{i=1}^n$, assuming a linear relationship between x_i and y_i:

$$y_i = x_i^T \beta + \epsilon_i \quad i = 1, \dots, n$$

where $x_i \in \mathbb{R}^p$ is the i-th input feature vector, $y_i \in \mathbb{R}$ is its corresponding output response and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is zero-mean Gaussian noise. If we stack the feature vectors x_i as rows in matrix $X \in \mathbb{R}^{n \times p}$ and stack the output responses y_i as a column vector $Y \in \mathbb{R}^n$, we can write the system of linear equations in matrix form:

$$Y = X\beta + \epsilon$$

where $\epsilon \in \mathbb{R}^n$ is a Gaussian noise vector with zero mean ($\mathbb{E}[\epsilon] = 0$) and a covariance matrix equal to a scaled identity matrix ($\mathbb{E}[\epsilon\epsilon^T] = \sigma^2 I_{n \times n}$). Throughout this script we assume X is full rank ($\text{rank}(X) = p$), which ensures that the matrix $X^T X$ is invertible. The optimal solution to a linear regression problem is called the **Ordinary Least Squares (OLS)** solution, which is stated in the following lemma:

Lemma 1. *(Ordinary Least Squares) Given a matrix $X \in \mathbb{R}^{n \times p}$ of input features and a vector $Y \in \mathbb{R}^n$ of observed responses, the Ordinary Least Squares solution to the regression problem $Y = X\beta + \epsilon$, where ϵ is a vector of i.i.d. random variables ϵ_i ($i = 1, \dots, n$) with $\mathbb{E}[\epsilon_i] = 0$ and $Var(\epsilon_i) = \sigma^2$, is defined as:*

$$\beta_{OLS} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \; \|Y - X\beta\|_2^2$$

The above minimization problem has solution $\beta_{OLS} = (X^TX)^{-1}X^TY$. The estimator is unbiased, $\mathbb{E}[\beta_{OLS}] = \beta$, and the Covariance matrix of the estimator is $\Sigma^{-1} = \mathbb{E}[(\beta - \beta_{OLS})(\beta - \beta_{OLS})^T] = \sigma^2(X^TX)^{-1}$.
(Note: We choose symbol Σ^{-1} for the covariance matrix as we will soon define an entity called the Information matrix with the symbol Σ, and we will show that for a linear model these two matrices are reciprocals.)

Proof. First we derive β_{OLS}. The objective $\|Y - X\beta\|_2^2$ is convex and differentiable w.r.t. β, meaning that we can find the minimizing β_{OLS} by finding β s.t. $\nabla_\beta\|Y - X\beta\|_2^2 = 0$:

$$\begin{aligned}\nabla_\beta\|Y - X\beta\|_2^2 &= 2X^T(Y - X\beta) = 0\\ \rightarrow X^TX\beta_{OLS} &= X^TY\\ \rightarrow \beta_{OLS} &= (X^TX)^{-1}X^TY\end{aligned}$$

Note that we assume X is full-rank (rank$(X) = p$) and thus, X^TX is invertible.
Next, we prove that the estimator is unbiased, *i.e.* $\mathbb{E}[\beta_{OLS}] = \beta$:

$$\begin{aligned}\mathbb{E}[\beta_{OLS}] &= \mathbb{E}[(X^TX)^{-1}X^TY]\\ &= \mathbb{E}[(X^TX)^{-1}X^T(X\beta + \epsilon)]\\ &= \mathbb{E}[(X^TX)^{-1}X^TX\beta + (X^TX)^{-1}X^T\epsilon]\\ &= \mathbb{E}[\beta] + \mathbb{E}[(X^TX)^{-1}X^T\epsilon]\\ &= \beta + (X^TX)^{-1}X^T\mathbb{E}[\epsilon] \qquad\qquad (\mathbb{E}[\epsilon] = 0)\\ &= \beta\end{aligned}$$

Finally, we prove that the covariance matrix $\Sigma^{-1} = \sigma^2(X^TX)^{-1}$. Starting from the definition of the covariance matrix ($\Sigma^{-1} = \mathbb{E}[(\beta - \beta_{OLS})(\beta - \beta_{OLS})^T]$), we have:

$$\begin{aligned}\Sigma^{-1} &= \mathbb{E}[(\beta_{OLS} - \beta)(\beta_{OLS} - \beta)^T]\\ &= \mathbb{E}[((X^TX)^{-1}X^TY - \beta)((X^TX)^{-1}X^TY - \beta)^T]\\ &= \mathbb{E}\big[((X^TX)^{-1})(X^TY - X^TX\beta)(X^TY - X^TX\beta)^T(X^TX)^{-1})\big]\\ &= \mathbb{E}\big[((X^TX)^{-1})(X^T\epsilon)(X^T\epsilon)^T(X^TX)^{-1})\big]\\ &= (X^TX)^{-1}X^T\mathbb{E}[\epsilon\epsilon^T]X(X^TX)^{-1}\\ &= (X^TX)^{-1}X^T\sigma^2IX(X^TX)^{-1}\\ &= \sigma^2(X^TX)^{-1}X^TX(X^TX)^{-1}\\ &= \sigma^2(X^TX)^{-1}\end{aligned}$$

□

The linear regression model has the property that the covariance matrix depends on neither the true parameter vector β nor the observed responses Y. This suggests that we can "optimize" the covariance of the estimator a priori, even before taking any measurements Y.

Next we define an entity closely related to the covariance matrix: Fisher Information (or Information Matrix).

Definition 1. *(Fisher Information) The Fisher Information is a measure of the amount of information that a vector of observable random variables $X = (X_1, ..., X_n)$ carries about an unknown parameter θ. Formally, it is the variance of the gradient (w.r.t θ) of the log-likelihood of X and in the multivariate case (where θ is a vector of parameters) the Information matrix $I(\theta)$ becomes:*

$$I(\theta)_{ij} = \mathbb{E}[\frac{\partial}{\partial\theta_i}\ \mathit{log}\ f(X;\theta) \times \frac{\partial}{\partial\theta_j}\ \mathit{log}\ f(X;\theta)]$$

Let us briefly derive the information matrix $I(\beta)$ for the linear regression model $Y = X\beta + \epsilon$. Since we defined each random noise variable ϵ_i as Gaussian ($\epsilon_i \sim \mathcal{N}(0, \sigma^2)$), it follows that the response variable y_i is also Gaussian ($y_i \sim \mathcal{N}(x_i^T\beta, \sigma^2)$). If $\{x_i, y_i\}_{i=1}^n$ are i.i.d, the log-likelihood $\log p(Y|X;\beta)$ can be written as:

$$\begin{aligned}\log p(Y|X;\beta) &= \log \prod_{i=1}^{n} p(y_i|x_i,\beta) = \sum_{i=1}^{n} \log(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - x_i\beta)^2}{2\sigma^2}}) \\ &= -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i\beta)^2\end{aligned}$$

The gradient of the log-likelihood (in matrix form) is $\nabla_\beta \log p(Y|X;\beta) = \frac{1}{\sigma^2}X^T(Y - X\beta)$. Using Definition 1, we can derive the information matrix $I(\beta)$ using the gradient $\nabla_\beta \log p(Y|X;\beta)$:

$$\begin{aligned}I(\beta) &= \mathbb{E}[\nabla_\beta \log p(Y|X;\beta) \times (\nabla_\beta \log p(Y|X;\beta))^T] \\ &= \mathbb{E}[\frac{1}{\sigma^2}X^T(Y - X\beta)(\frac{1}{\sigma^2}X^T(Y - X\beta))^T] \\ &= \mathbb{E}[\frac{1}{\sigma^4}X^T\epsilon(X^T\epsilon)^T] \quad (\epsilon = Y - X\beta) \\ &= \mathbb{E}[\frac{1}{\sigma^4}X^T\epsilon\epsilon^T X] = \frac{1}{\sigma^2}X^T X \quad (\mathbb{E}[\epsilon\epsilon^T] = \sigma^2 I_{n\times n})\end{aligned}$$

Hence, $I(\beta)$ is exactly the inverse of the covariance matrix Σ^{-1}. From now on, we denote the information matrix Σ, and minimizing the covariance matrix Σ^{-1} is identical to maximizing Σ. Using the information matrix $\Sigma = \frac{1}{\sigma^2}X^T X$, we can reason geometrically about how accurate the parameter vector β_{OLS} is estimated compared to the true (unknown) parameter vector β. The α-confidence level ellipsoid for β is defined as:

$$\mathcal{E} = \{z|(z - \beta_{OLS})^T\Sigma(z - \beta_{OLS}) \leq \kappa\}$$

where κ is a constant that depends on α. This p-dimensional ellipsoid encloses, with confidence α, all vectors z that could be the true parameter vector β. As described in [2], the ellipsoid can be visualized by doing Singular-value decomposition of X. If $X = U\Lambda V$, where Λ is the matrix of singular values and U and V are the left- and right singular vectors, the ellipsoid axis directions are given by V and the lengths are inversely proportional to their respective singular values, as illustrated in Figure 1. It is easy to see that maximizing the singular values λ_1, λ_2 minimizes the ellipsoid and thus constrains β to be close to β_{OLS}.

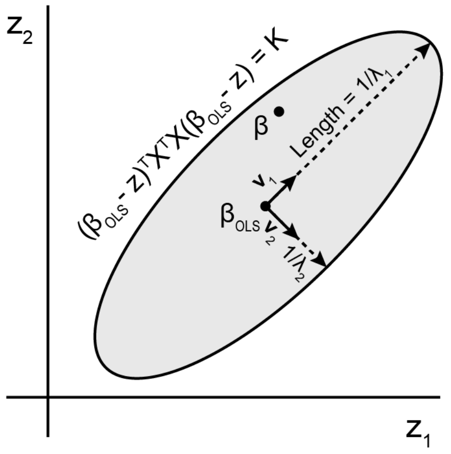

Figure 1: Confidence ellipsoid around the estimated parameter vector β_{OLS} (in two dimensions). The axis directions are given by the right singular vectors v_1, v_2 of X and their lengths are inversely proportional to the singular values λ_1, λ_2. With confidence level α (α determines K of the ellipsoid boundary condition), the true parameter vector β resides inside the ellipsoid.

1.2 Problem statement: Optimal Design

In linear experimental design, we assume a linear model $Y = X\beta + \epsilon$ (as defined in the previous section), where $X \in \mathbb{R}^{n \times p}$ is a matrix of n design points (the input feature vectors), $Y \in \mathbb{R}^n$ is the vector of observable responses and $\epsilon \in \mathbb{R}^n$ is an i.i.d Gaussian noise vector with zero mean and finite variance. The design problem refers to selecting a small subset $S \subset \{1, ..., n\}$ of r rows, X_S, from X so that the precision of estimating β is maximized on the selected design X_S. In classical experimental design, we associate a cost with conducting each experiment (row of X). By selecting r rows of X, where $r << n$, that are most statistically efficient for estimating β, the total experimental cost is minimized. When n is very large, optimal design is a strategy for doing regression efficiently, as it might be computationally intractable to perform OLS on the entire dataset.

Since we are looking for S such that X_S is most statistically efficient, the optimal design problem reduces to minimizing the covariance matrix $\Sigma^{-1} = (X_S^T X_S)^{-1}$. A number of summary statistics, called *Optimality criteria*, have been developed for measuring how well Σ^{-1} is minimized on a selected design X_S. Usually, an optimality criterion is a function $f : \mathcal{S}_p^+ \to \mathbb{R}$ (where $\mathcal{S}_p^+$ is the set of $p \times p$-dimensional positive-definite matrices) that maps Σ^{-1} to a real number. The experimental design problem can then be formulated as the following optimization problem (as defined in [3]):

$$S^*(r) = \arg\min_S f(X_S^T X_S)$$

where S is a set or multi-set of size r, depending on how the the design problem is constrained:

1. **With replacement**: S is a multi-set of row subscripts of X such that $|S| = r$.
   Under this setting, X_S may contain duplicate rows of the matrix X.

2. **Without replacement**: S is a subset of row subscripts of X ($S \subset [n]$) such that $|S| = r$.
   In this case, X_S can only contain distinct rows of the matrix X.

By replicating each row of X r times, the "Without replacement" setting reduces to the "With replacement" setting. As in [4], the "With replacement" optimization problem seeks Σ^{-1} where:

$$\textit{minimize } (w.r.t\ S_p^+)\ \Sigma^{-1} = (\sum_{j=1}^{n} m_j x_j x_j^T)^{-1}$$

$$\textit{subject to } m_i \geq 0,\ m_1 + \cdots + m_n = r,\ m_i \in \mathbb{Z}$$

The above optimization problem is NP-hard [3]. If we drop the condition that the relative frequency of a certain measurement should be an integer multiple of $\frac{1}{k}$, we arrive at the following relaxed problem:

$$\textit{minimize } (w.r.t\ S_p^+)\ \Sigma^{-1} = \frac{1}{r}(\sum_{j=1}^{n} \lambda_j x_j x_j^T)^{-1}$$

$$\textit{subject to } \lambda_i \geq 0,\ \sum_{j=1}^{n} \lambda_j = 1$$

Having expressed experimental design as a minimization problem of an optimality criterion $f(\Sigma)$, we now turn to defining some of the most widely used optimality criteria f, summarized from [5]. We also reason about the intuition of some of these criteria graphically in the context of the confidence ellipsoid.

- **A-optimality (Average)**: $f_A(\Sigma) = \frac{1}{p} Tr(\Sigma^{-1})$
  The objective is simply the mean of the norm of the squared error:

$$\mathbb{E}[||e||_2^2] = \mathbb{E}[Tr(ee^T)] = Tr(\mathbb{E}[ee^T]) = Tr(\Sigma^{-1})$$

It thus minimizes the average variance of the estimated regression coefficients β. It is computationally very appealing since it only involves computing the diagonal of Σ^{-1}. Figure 2 (left) illustrates the effect of A-Optimality on the resulting confidence ellipsoid around β_{OLS}. Minimizing the average trace of $(X^TX)^{-1}$ is identical to minimizing the average of the inverse singular values $\frac{1}{\lambda_i}$ of X. Thus, A-optimality minimizes the average axis length of the ellipsoid.

- **D-optimality (Determinant)**: $f_D(\Sigma) = (det|\Sigma|)^{-\frac{1}{p}}$
  This criterion minimizes the determinant of Σ^{-1}, which according to [5] is proportional to the volume of the confidence ellipsoid (for a fixed confidence level). Thus, as illustrated in Figure 2 (center), D-optimality shrinks the ellipsoid in all directions in order to minimize total volume.

- **T-optimality (Trace)**: $f_T(\Sigma) = \frac{p}{Tr(\Sigma)}$
  This criterion is similar to A-optimality, as it minimizes the sum of variances. Here, however, the trace is applied to the information Σ instead of Σ^{-1}. While T-optimality is suitable for some domains, [5] shows that it behaves poorly in certain circumstances. For example, if all input vectors x_i are of constant squared norm, $Tr(X^TX) = \sum_i \|x_i\|_2^2$ is a constant and hence useless for assessing designs.

- **E-optimality (Eigenvalue)**: $f_E(\Sigma) = ||\Sigma^{-1}||_2$
  This criterion minimizes the maximum eigenvalue of Σ^{-1}. Since the diameter (twice the longest semi-axis) of the confidence ellipsoid is proportional to the square root of the maximum eigenvalue, minimizing this quantity is geometrically interpreted as minimizing the diameter, as illustrated in Figure 2 (right). Alternatively, E-optimal design can be interpreted as minimizing the maximum variance of q^Te over all q with unit norm, guarding against the worst possible variance.

- **V-optimality (Variance)**: $f_V(\Sigma) = \frac{1}{n}Tr(X\Sigma^{-1}X^T)$
  The criterion can be re-formulated as:

$$\frac{1}{n}Tr(X\mathbb{E}[ee^T]X^T) = \frac{1}{n}\mathbb{E}[Tr(e^TX^TXe)] = \frac{1}{n}\mathbb{E}[||Xe||_2^2]$$

  Hence, it minimizes the average prediction variance, *i.e.* the variance of the prediction variable Y.

- **G-optimality**: $f_G(\Sigma) = \max \operatorname{diag}(X\Sigma^{-1}X^T)$
  G-optimality minimizes the largest diagonal entry of the projection matrix, which minimizes the worst possible prediction variance (in contrast to V-optimality that minimizes average prediction variance).

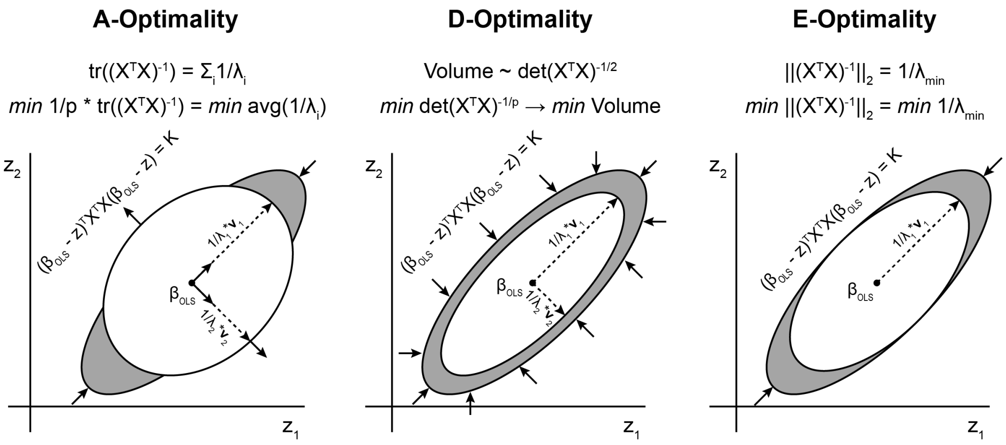
A-Optimality
$tr((X^TX)^{-1}) = \Sigma_i 1/\lambda_i$
min $1/p * tr((X^TX)^{-1})$ = *min* $avg(1/\lambda_i)$
D-Optimality
Volume ~ $det(X^TX)^{-1/2}$
min $det(X^TX)^{-1/p} \rightarrow$ *min* Volume
E-Optimality
$||(X^TX)^{-1}||_2 = 1/\lambda_{min}$
min $||(X^TX)^{-1}||_2$ = *min* $1/\lambda_{min}$
$(\beta_{OLS} - z)^TX^TX(\beta_{OLS} - z) = K$
$1/\lambda_1 * v_1$
$1/\lambda_2 * v_2$
β_{OLS}
z_1
z_2

Figure 2: The effects of the Optimality-criteria A, D, and E on the confidence ellipsoid around β_{OLS}. The arrows surrounding the gray area illustrate how the ellipsoid shrinks due to optimizing for a criterion.

All criteria above are termed regular, which means they satisfy two conditions:
First, they are all convex:

$$f(\lambda A+(1-\lambda)B)\le \lambda f(A)+(1-\lambda)f(B), \text{ for all positive-definite matrices A, B and } 0\le\lambda\le 1$$

Second, they all satisfy reciprocal multiplicity:

$$f(tA)=t^{-1}f(A), \text{ for all positive-definite matrices A and } t>0$$

1.3 Optimal Design via Randomized Sampling

We will now explore Linear experimental design in the context of Randomized sampling, as regarded in [1]. Consider a large-scale least-squares problem. Given the full dataset $(X,Y)\in\mathbb{R}^{n\times p}\times\mathbb{R}^n$, sketching algorithms use a sketching matrix $S\in\mathbb{R}^{r\times n}$ with $r\ll n$ to construct a reduced, "sketched" version $(SX,\ SY)$ of the data (*i.e.* sketching algorithms project (X,Y) to lower-dimensional space by multiplication of S). A random sampling sketch matrix S has exactly one non-zero entry at each row. Formally, sampling matrix S is formed by the matrix product $\tilde{S}W$, where $\tilde{S}$ and W are defined as follows:

$$\tilde{S}\in\mathbb{R}^{r\times n} \text{ is a non-scaled random sampling matrix, where } \tilde{S}_{ij}\in\{0,1\} \text{ and } \sum_{j=1}^{n}\tilde{S}_{ij}=1$$

$$W\in\mathbb{R}^{n\times n} \text{ is a diagonal matrix of scaling factors } W_{jj}>0,\ \forall j$$

Thus, a sketch constructed from a random sampling matrix S consists of r typically re-scaled rows of (X,Y). Sketching algorithms, similar to optimal design, are concerned with assuring that the sketched estimator of β is not much worse than the original Least-Squares (LS) estimator. The objective is to approximate the LS solution of dataset (X,Y) by solving the LS problem of $(SX,\ SY)$:

$$\beta_S=\underset{\beta\in\mathbb{R}^p}{\arg\min}\|SY-SX\beta\|_2^2$$

Since $r<<n$, solving the LS problem for the sketched data has computational complexity $O(rp^2)$, which is significantly less compared to the original LS problem ($O(np^2)$). This is the main motivation for studying sketching methods in [1]. But we can also draw parallels to optimal design: The sampling-based sketching is identical to picking r design points in a Linear experiment that are most statistically efficient for estimating the parameters. Assuming that $rank(X)=p$, the OLS solution for β is (See Lemma 1):

$$\beta_{OLS}=(X^TX)^{-1}X^TY$$

Similarly, the solution β_S of the sketched LS problem is:

$$\beta_S=((SX)^TSX)^{-1}(SX)^TSY$$

As we did in Lemma 1 for OLS, we can derive the covariance matrix of the sketched estimator as:

$$\Sigma^{-1}=\sigma^2((SX)^TSX)^{-1}$$

The optimality of sketching matrices is assessed in [1] using the two statistical criteria shown below.

Definition 2. *(Criterion 1: Prediction Efficiency) The first criterion evaluates S according to*

$$C_{PE}(S)=\frac{\mathbb{E}[||X(\beta-\beta_S)||_2^2]}{\mathbb{E}[||X(\beta-\beta_{OLS})||_2^2]}$$

Definition 3. *(Criterion 2: Residual Efficiency) The second criterion evaluates S according to*

$$C_{RE}(S)=\frac{\mathbb{E}[||Y-X\beta_S||_2^2]}{\mathbb{E}[||Y-X\beta_{OLS}||_2^2]}$$

The two criteria look similar, but there are important differences: C_{PE} (Definition 2) measures the ratio between the expected true prediction error of β_S and the expected true prediction error of β_{OLS}. In contrast, C_{RE} (Definition 3) measures the ratio between the expected residual errors. The latter quantity should intuitively be easier to minimize, as it compares the estimator to the trained-on, noisy, responses Y, while C_{PE} compares the estimator to the optimal solution derived from OLS on the full dataset.

The sampling-based sketching algorithms rely on entities called "Leverage scores", which we define below [6]:

Definition 4. *(Leverage Scores) For matrix X, the statistical leverage scores are defined as follows: Let $X = U\Lambda V^T$ be the singular value decomposition (SVD) for matrix X. Let U denote the $n \times p$ matrix consisting of the p left singular vectors of X and let $U_{(i)}$ denote its i-th row as a row vector. Then, the statistical leverage scores of the rows of X, for $i = 1, \dots, n$, are given by:*

$$\ell_i = ||U_{(i)}||_2^2$$

The coherence γ of the rows of X is defined as:

$$\gamma = \max_{i \in \{1,\dots,n\}} \ell_i$$

If $X = U\Lambda V^T$ is the SVD of X, we can express the projection matrix $P_X = X(X^TX)^{-1}X^T$ (which projects the observed response vector Y into the prediction vector $\hat{Y} = X(X^TX)^{-1}X^TY = X\beta_{OLS}$) as:

$$\begin{aligned} X(X^TX)^{-1}X^T &= U\Lambda V^T(V\Lambda U^TU\Lambda V^T)^{-1}V\Lambda U^T \\ &= U\Lambda V^TV\Lambda^{-2}V^TV\Lambda U^T \\ &= U\Lambda^{-1}\Lambda U^T \\ &= UU^T \end{aligned}$$

Therefore, we can write the statistical leverage scores as follows:

$$\ell_i = ||U_{(i)}||_2^2 = (UU^T)_{ii} = (X(X^TX)^{-1}X^T)_{ii} = (P_X)_{ii}$$

where P_X is the projection matrix. Each leverage score corresponds to one row of X and the higher the leverage score is, the more influence the data point has on the estimation. Also note that:

$$\sum_{i=1}^{n} \ell_i = Tr(UU^T) = Tr(U^TU) = Tr(I_{p\times p}) = p$$

Now, we present two random sampling-based sketching algorithms from [1].

Algorithm 1. *(Random Sampling with Rescaling)*

1. *Construct a random sampling matrix $\tilde{S} \in \mathbb{R}^{r\times n}$ s.t. $\tilde{S}_{ij} \in \{0,1\}$ and $\sum_{j=1}^{n} \tilde{S}_{ij} = 1$. Generate each row of $\tilde{S}$ from an independent distribution $(p_i)_{i=1}^n$, where*
   - *$p_i = (1-\theta)\frac{\ell_i}{p} + \theta q_i$ and q_i is an arbitrary probability distribution,*
   - *$\ell_i = (UU^T)_{ii}$ denotes the leverage score of the i-th example (when $X = U\Lambda V^T$ is the SVD of X),*
   - *$\theta \in [0,1]$ is the mixing weight of q_i.*
2. *Construct a diagonal scaling matrix $W \in \mathbb{R}^{n\times n}$ s.t. $W_{ii} = \sqrt{\frac{1}{rp_i}}$.*
3. *Return the final sketching matrix $S_R = \tilde{S}W$.*

The next algorithm is a special case of Algorithm 1, where we do not re-scale the sampling matrix.

Algorithm 2. *(Random Sampling without Rescaling)*

1. *Construct the random sampling matrix* $\tilde{S} \in \mathbb{R}^{r \times n}$ *from Step 1 of Algorithm 1, setting* $\theta = 0$.
2. *Return the sketching matrix* $S_{NR} = \tilde{S}$ *(without re-scaling it, as was done in Step 2 of Algorithm 1).*

Re-scaling of sketching matrices are done to ensure the estimator is unbiased, however we will see later in this section that under certain assumptions, no rescaling can lead to even better bounds on estimator variance.

In order to show that sketching matrices constructed according to leverage scores of X result in an accurate estimator, let us first examine the properties of the Prediction- and Residual efficiencies (Definition 2 & 3).

Lemma 2. *(Structural properties of the Sketched estimator)*
Given that sketching matrix S preserves rank, i.e. rank(SX) = rank(SU) = p, the following properties hold:

$$\begin{aligned}
C_{PE}(S) &= \frac{\mathbb{E}[||X(\beta - \beta_S)||_2^2]}{\mathbb{E}[||X(\beta - \beta_{OLS})||_2^2]} = \frac{\|U((SU)^T SU)^{-1}(SU)^T S\|_F^2}{p} \\
C_{RE}(S) &= \frac{\mathbb{E}[||Y - X\beta_S||_2^2]}{\mathbb{E}[||Y - X\beta_{OLS}||_2^2]} = 1 + \frac{\|U((SU)^T SU)^{-1}(SU)^T S\|_F^2 - 1}{n/p - 1}
\end{aligned}$$

Proof. First, consider the expected prediction error $\mathbb{E}[\|X(\beta_{OLS} - \beta)\|_2^2]$ when solving OLS using the full data set (X, Y). Given the SVD of X as $X = U\Lambda V^T$, the expectation can be re-formulated as:

$$\begin{aligned}
\mathbb{E}[\|X(\beta_{OLS} - \beta)\|_2^2] &= \mathbb{E}[\|X\beta_{OLS} - X\beta\|_2^2] \\
&= \mathbb{E}[\|X\beta_{OLS} - U\Lambda V^T\beta\|_2^2] \qquad (X = U\Lambda V^T) \\
&= \mathbb{E}[\|UU^TY - U\Lambda V^T\beta\|_2^2] \qquad (X\beta_{OLS} = X(X^TX)^{-1}X^TY = UU^TY) \\
&= \mathbb{E}[\|UU^TU\Lambda V^T\beta + UU^T\epsilon - U\Lambda V^T\beta\|_2^2] \quad (\text{Linear model } Y = U\Lambda V^T\beta + \epsilon) \\
&= \mathbb{E}[\|UU^T\epsilon\|_2^2] \qquad (U^TU = I_{p\times p}) \\
&= \sigma^2 p \qquad (\mathbb{E}[\epsilon\epsilon^T] = \sigma^2 I_{p \text{ x } p} \text{ and } \|UU^T\|_2^2 = p)
\end{aligned}$$

Now, let us derive the Expected prediction error $\mathbb{E}[\|X(\beta_S - \beta)\|_2^2]$, having applied sketching matrix S to the data before solving the LS problem (obtaining solution β_S):

$$\begin{aligned}
&\mathbb{E}[\|X(\beta_S - \beta)\|_2^2] = \mathbb{E}[\|X\beta_S - X\beta\|_2^2] \\
&\quad = \mathbb{E}[\|X\beta_S - U\Lambda V^T\beta\|_2^2] \qquad (X = U\Lambda V^T) \\
&\quad = \mathbb{E}[\|U((SU)^TSU)^{-1}(SU)^TSY - U\Lambda V^T\beta\|_2^2] \qquad (X\beta_S = U((SU)^TSU)^{-1}(SU)^TSY) \\
&\quad = \mathbb{E}[\|U((SU)^TSU)^{-1}(SU)^TS\epsilon\|_2^2] + \mathbb{E}[\|(((SU)^TSU)^{-1}(SU)^TSU - I_{p \text{ x } p})U\Lambda V^T\beta\|_2^2] \quad (Y = U\Lambda V^T\beta + \epsilon) \\
&\quad = \sigma^2\|U((SU)^TSU)^{-1}(SU)^TS\|_F^2 \quad (\text{Pulling out } \epsilon \text{ from the vector norm gives a } \|\cdot\|_F \text{ norm})
\end{aligned}$$

Finally, dividing $\mathbb{E}[\|X(\beta_S - \beta)\|_2^2]$ by $\mathbb{E}[\|X(\beta_{OLS} - \beta)\|_2^2]$ completes the proof of the lemma for $C_{PE}(S)$. A similar derivation for $C_{RE}(S)$ proceeds below, keeping in mind that $Y = U\Lambda V^T\beta + \epsilon$ and $X\beta_{OLS} = UU^TY$:

$$\begin{aligned}
\mathbb{E}[\|Y - X\beta_{OLS}\|_2^2] &= \mathbb{E}[\|U\Lambda V^T\beta + \epsilon - UU^TU\Lambda V^T\beta - UU^T\epsilon\|_2^2] \\
&= \mathbb{E}[\|(I_{n \text{ x } n} - UU^T)\epsilon\|_2^2] \\
&= \sigma^2\|(I_{n \text{ x } n} - UU^T)\|_F^2 = \sigma^2(n - p)
\end{aligned}$$

Given that $X\beta_S = X((SX)^T(SX))^{-1}(SX)^TY = U((SU)^TSU)^{-1}(SU)^TSY$, we have:

$$\begin{aligned}\mathbb{E}[\|Y - X\beta_S\|_2^2] &= \mathbb{E}[\|U\Lambda V^T\beta + \epsilon - U((SU)^TSU)^{-1}(SU)^TSU\Lambda V^T\beta - U((SU)^TSU)^{-1}(SU)^TS\epsilon\|_2^2] \\ &= \mathbb{E}[\|\epsilon - U((SU)^TSU)^{-1}(SU)^TS\epsilon\|_2^2] \\ &= \mathbb{E}[\|(I_{n \text{ x } n} - U((SU)^TSU)^{-1}(SU)^TS)\epsilon\|_2^2] \\ &= \sigma^2 Tr((I_{n \text{ x } n} - U((SU)^TSU)^{-1}(SU)^TS)^T(I_{n \text{ x } n} - U((SU)^TSU)^{-1}(SU)^TS)) \\ &= \sigma^2(Tr(I_{n \text{ x } n}) - 2Tr(U((SU)^TSU)^{-1}(SU)^TS) + \|U((SU)^TSU)^{-1}(SU)^TS\|_F^2) \\ &= (n - 2p + \|U((SU)^TSU)^{-1}(SU)^TS\|_F^2)\sigma^2\end{aligned}$$

As before, dividing $\mathbb{E}[\|Y - X\beta_S\|_2^2]$ by $\mathbb{E}[\|Y - X\beta_{OLS}\|_2^2]$ concludes the proof of the lemma.

□

Clearly, we would like to bound $\mathbb{E}[\|X(\beta_S - \beta)\|_2^2]$ and $\mathbb{E}[||Y - X\beta_S||_2^2]$ in such a way that they cannot be much larger than $\mathbb{E}[\|X(\beta_{OLS} - \beta)\|_2^2]$ and $\mathbb{E}[||Y - X\beta_{OLS}||_2^2]$. Lemma 2 shows that in order to bound these terms, we need to carefully choose sketching matrix S such that $\mathbb{E}[\|U((SU)^TSU)^{-1}(SU)^TS\|_F^2]$ is minimal. Note that this quantity looks very similar to the projection matrix $X(X^TX)^{-1}X^T$. Indeed, using the identity $X = U\Lambda V^T$, one can show that $U((SU)^TSU)^{-1}(SU)^TS = X((SX)^TSX)^{-1}(SX)^TS$. This is the projection matrix, constructed from the sampled dataset (SX, SY), that projects the full response vector Y into the prediction vector $\hat{Y}$ ($\hat{Y} = X((SX)^TSX)^{-1}(SX)^TSY = X\beta_S$). Being a projection matrix, we could apply the Optimal design criteria defined in Section 1.2 to minimize some statistical measure on it. Specifically, recall the V-optimality criterion $f_V(X_S) = \frac{1}{n}Tr(X(X_S^TX_S)^{-1}X^T)$ which minimizes (w.r.t the design matrix X_S) the average diagonal entry of the projection matrix. We argued that this is equivalent to minimizing the average prediction variance. If we minimized the trace of the projection matrix $U((SU)^TSU)^{-1}(SU)^TS$ (w.r.t S), by definition it is identical to minimizing the V-optimality criterion. Of course, this is an NP-hard problem [3]. Instead, by sampling S according to leverage scores $\ell_i = (UU^T)_{ii} = (X(X^TX)^{-1}X^T)_{ii}$ (as in Algorithm 1), one can achieve an upper bound with high probability, as is stated and proven in Theorem 2 below. First, we present an important auxiliary theorem required for the proof of Theorem 2. We will not prove the theorem here, but the interested reader may find a detailed proof in [7] (Theorem 4).

Theorem 1. *(Auxiliary theorem: Sampling matrix approximation error bound)*
Let $X = U\Lambda V^T$ be the SVD of matrix X, where $U \in \mathbb{R}^{n\times p}$ is the matrix of left singular vectors. Construct the scaled sampling matrix $S \in \mathbb{R}^{r\times n}$ exactly according to Algorithm 1 (with X as input). Let $\epsilon \in (0,1)$ be an accuracy parameter and let $\theta \in [0,1]$ be a hyper-parameter. If the number of sampled rows r satisfies

$$r \geq \frac{96p}{(1-\theta)\epsilon^2}\log\left(\frac{96p}{(1-\theta)\epsilon^2\sqrt{\delta}}\right)$$

then, with probability at least $1-\delta$,

$$\|U^TU - (SU)^TSU\|_2 \leq \epsilon$$

Note that since $U^TU = I_{p\times p}$, the inequality can be written as $\|I_{p\times p} - (SU)^TSU\|_2 \leq \epsilon$.
Further note that $\mathbb{E}[S^TS] = I_{n\times n}$.

Intuitively, the theorem states that the approximation error of any sampled orthogonal matrix can be bounded with high probability given that we sample enough rows. We are now ready to state Theorem 2, which bounds the statistical efficiency criteria (Definition 2 and 3) for Algorithm 1 with high probability.

Theorem 2. *(Upper bound: Random Sampling with Rescaling)*
If sampling matrix S_R is constructed from Algorithm 1 with $r \geq \frac{Cp}{1-\theta}log(\frac{C'p}{(1-\theta)\sqrt{\delta}})$, where $C, C' > 0$ are constants, then with probability $\geq (1-\delta)^2$:

$$C_{PE}(S_R) \le \frac{4}{\delta}(1 + \frac{n}{(1-\theta)r})$$
$$C_{RE}(S_R) \le 1 + \frac{4}{\delta}(\frac{p}{n} + \frac{p}{(1-\theta)r})$$

Proof. First, define the SVD of $SU = \tilde{U}\tilde{\Lambda}\tilde{V}^T$, *i.e.*, the singular-value decomposition of the sampled left singular matrix of X. Now, begin by substituting this decomposition for certain terms in C_{PE}:

$$\begin{aligned}\frac{\mathbb{E}[||X(\beta - \beta_S)||_2^2]}{\mathbb{E}[||X(\beta - \beta_{OLS})||_2^2]} &= \frac{1}{p}\|U((SU)^T SU)^{-1}(SU)^T S\|_F^2 \text{ (From the proof of Lemma 2)} \\ &= \frac{1}{p}\|(U^T S^T SU)^{-1} U^T S^T S\|_F^2 \\ &= \frac{1}{p}\|(\tilde{V}\tilde{\Lambda}\tilde{U}^T\tilde{U}\tilde{\Lambda}\tilde{V}^T)^{-1} U^T S^T S\|_F^2 \\ &= \frac{1}{p}\|\tilde{V}\tilde{\Lambda}^{-2}\tilde{V}^T U^T S^T S\|_F^2\end{aligned}$$

Observe the term $\tilde{V}\tilde{\Lambda}^{-2}\tilde{V}^T$. It is a $p \times p$ matrix where every element is less than or equal to the squared reciprocal of the smallest singular value α of the matrix SU. Hence, it follows that:

$$\frac{1}{p}\|\tilde{V}\tilde{\Lambda}^{-2}\tilde{V}^T U^T S^T S\|_F^2 \le \frac{1}{p}||\tilde{V}\tilde{\Lambda}^{-2}\tilde{V}^T||_2^2||U^T S^T S||_F^2 \le \frac{1}{\alpha^4}\frac{1}{p}\|U^T S^T S\|_F^2$$

First, to find an upper-bound for the quantity $\frac{1}{\alpha^4}$, we apply Theorem 1. The theorem states that if matrix S is constructed by sampling and re-scaling from U according to Algorithm 1, then $\|U^T U - (SU)^T SU\|_2 \le \epsilon$ with probability at least $1 - \delta$ if the number of sampled rows $r \ge \frac{96p}{(1-\theta)\epsilon^2}\ln(\frac{96p}{(1-\theta)\epsilon^2\sqrt{\delta}})$. Now, the 2-norm is defined as the maximum singular value of a matrix, and U is an orthonormal matrix, so its maximum singular value is exactly 1. So by setting $\epsilon = \frac{1}{\sqrt{2}}$, the squared maximum singular value $\frac{1}{\alpha^4} \le 4$ with probability $1 - \delta$. Let us now bound $\frac{1}{p}\mathbb{E}[\|U^T S^T S\|_F^2]$ (in order to apply Markov's inequality later):

$$\begin{aligned}\frac{1}{p}\mathbb{E}[\|U^T S^T S\|_F^2] &= \frac{1}{p}\mathbb{E}[Tr(U^T S^T S(U^T S^T S)^H)] \qquad (\|A\|_F^2 = Tr(AA^H) \text{ where } A^H \text{ is the Hermitian transpose}) \\ &= \frac{1}{p}\mathbb{E}[Tr(U^T S^T S(S^T S)^H (U^T)^H)] \qquad (U \text{ and } S^T S \text{ are unitary }) \\ &= \frac{1}{p}\mathbb{E}[Tr(U^T (S^T S)^2 U)] \\ &= \frac{1}{p}\mathbb{E}[\sum_{j=1}^{p}\sum_{i=1}^{n}\sum_{k=1}^{n} U_{ij}U_{kj}(S^T S)_{ki}^2] \qquad ((S^T S)_{ki} = 0 \text{ if } k \neq i) \\ &= \frac{1}{p}\mathbb{E}[\sum_{j=1}^{p}\sum_{i=1}^{n} U_{ij}^2 (S^T S)_{ii}^2] = \frac{1}{p}\mathbb{E}[\sum_{i=1}^{n} \ell_i (S^T S)_{ii}^2] \qquad (\sum_{j=1}^{p} U_{ij}^2 = \|u_i\|_2^2 = \ell_i)\end{aligned}$$

The term we want to bound is thus proportional to the leverage scores ℓ_i of X and the diagonal of squares of our sketching matrix S. Remember that in Algorithm 1 we sample each element S_{ki} of S according to $S_{ki} = \frac{1}{\sqrt{rp_i}}\sigma_{ki}$. Here the row index $k \in \{1, ..., r\}$ denotes each independent sample in S and the column index $i \in \{1, ..., n\}$ denotes the sampled row from X and $\frac{1}{\sqrt{rp_i}}$ is a scaling factor. $\sigma_{ki} \in \{0, 1\}$ is a random variable and $\sigma_{ki} = 1$ means that for sample instance k, we selected row i from X. Also, remember that the sampling probability for a single σ_{ki} is $P(\sigma_{ki} = 1) = p_i = (1 - \theta)\frac{\ell_i}{p} + \theta q_i$. Substituting these expressions, we get:

$$\begin{aligned}\frac{1}{p}\mathbb{E}[\sum_{i=1}^{n}\ell_i(S^TS)^2_{ii}] &= \frac{1}{r^2p}\sum_{i=1}^{n}\frac{\ell_i}{p_i^2}\sum_{m=1}^{r}\sum_{l=1}^{r}\mathbb{E}[\sigma_{mi}^2\sigma_{li}^2] \qquad (\sigma_{ab}^2 = \sigma_{ab} \in \{0,1\})\\ &= \frac{1}{r^2p}\sum_{i=1}^{n}\frac{\ell_i}{p_i^2}\sum_{m=1}^{r}\sum_{l=1}^{r}\mathbb{E}[\sigma_{mi}\sigma_{li}]\end{aligned}$$

To decompose this expression, observe the following: When constructing sampling matrix S, we create each sample instance (*i.e.* row in S) independently. Then, for any two rows a and b of S where $a \neq b$, σ_{ai} and σ_{bi} are independent and identical (they have identical probabilities p_i for sampling row i of X), which means $\mathbb{E}[\sigma_{ai}\sigma_{bi}] = \mathbb{E}[\sigma_{ai}]\mathbb{E}[\sigma_{bi}] = p_i^2$. In the sum over $m = 1$ to r and $l = 1$ to r, there are $r^2 - r$ such independent elements (all except the diagonal). For the r elements where $a = b$, $\mathbb{E}[\sigma_{ai}\sigma_{ai}] = \mathbb{E}[\sigma_{ai}^2] = \mathbb{E}[\sigma_{ai}] = p_i$. Hence:

$$\begin{aligned}\frac{1}{r^2p}\sum_{i=1}^{n}\frac{\ell_i}{p_i^2}\sum_{m=1}^{r}\sum_{l=1}^{r}\mathbb{E}[\sigma_{mi}\sigma_{li}] &= \frac{1}{r^2p}\sum_{i=1}^{n}\frac{\ell_i}{p_i^2}((r^2-r)p_i^2 + rp_i)\\ &= \frac{1}{r^2p}\sum_{i=1}^{n}((\ell_i(r^2-r) + r\frac{\ell_i}{p_i}))\\ &= 1 - \frac{1}{r} + \frac{1}{rp}\sum_{i=1}^{n}\frac{\ell_i}{p_i}\end{aligned}$$

Even clearer now, we can bound this expectation by just sampling p_i proportional to ℓ_i. Substituting $p_i = (1-\theta)\frac{\ell_i}{p} + \theta q_i$ yields the final bound:

$$\begin{aligned}1 - \frac{1}{r} + \frac{1}{rp}\sum_{i=1}^{n}\frac{\ell_i}{p_i} &= 1 - \frac{1}{r} + \frac{1}{rp}\sum_{i=1}^{n}\frac{\ell_i}{(1-\theta)\frac{\ell_i}{p} + \theta q_i}\\ &\leq 1 - \frac{1}{r} + \frac{n}{(1-\theta)r}\\ &\leq 1 + \frac{n}{(1-\theta)r}\end{aligned}$$

Which implies that:

$$\frac{1}{p}\mathbb{E}[\|U^TS^TS\|_F^2] \leq 1 + \frac{n}{(1-\theta)r}$$

Markov's Inequality states that if $X \geq 0$ is a random variable, then for $a > 0$, $P(X > a) \leq \frac{\mathbb{E}[X]}{a}$. Hence, by setting $X = \frac{1}{p}\|U^TS^TS\|_F^2$ and $a = \frac{1}{\delta}\frac{1}{p}\mathbb{E}[\|U^TS^TS\|_F^2]$, we get:

$$P\left(\frac{1}{p}\|U^TS^TS\|_F^2 > \frac{1}{\delta}\frac{1}{p}\mathbb{E}[\|U^TS^TS\|_F^2]\right) \leq \delta$$

Which implies that:

$$\frac{1}{p}\|U^TS^TS\|_F^2 \leq \frac{1}{\delta}(1 + \frac{n}{(1-\theta)r}) \text{ with probability } \geq 1 - \delta$$

Putting it together, we need both $\frac{1}{\alpha^4} \leq 4$ and $\frac{1}{p}\|U^T S^T S\|_F^2 \leq \frac{1}{\delta}(1 + \frac{n}{(1-\theta)r})$ to hold (and each bound holds with probability $1-\delta$). Hence:

$$\begin{aligned} C_{PE}(S_R) &= \frac{\|U((SU)^T SU)^{-1}(SU)^T S\|_F^2}{p} \\ &\leq \frac{1}{\alpha^4}\frac{1}{p}\|U^T S^T S\|_F^2 \\ &\leq 4 \times \frac{1}{\delta}(1 + \frac{n}{(1-\theta)r}) \\ &\leq \frac{4}{\delta}(1 + \frac{n}{(1-\theta)r}) \quad \text{with probability } \geq (1-\delta)^2 \end{aligned}$$

Because C_{RE} contains the same terms as C_{PE}, just scaled by a factor, its proof is identical, and we get:

$$\begin{aligned} C_{RE}(S_R) &= 1 + \frac{\|U((SU)^T SU)^{-1}(SU)^T S\|_F^2 - 1}{n/p - 1} \\ &\leq 1 + \frac{p}{n}C_{PE}(S_R) \\ &\leq 1 + \frac{4}{\delta}(\frac{p}{n} + \frac{p}{(1-\theta)r}) \text{ with probability } \geq (1-\delta)^2 \end{aligned}$$

□

To summarize the two bounds, the Residual Efficiency (C_{RE}) scales as $\frac{p}{r}$, while the Prediction Efficiency (C_{PE}) scales as $\frac{n}{r}$. In other words, by setting the number of design points r to be proportional to p makes the Sketching approximation bounded by a constant factor for C_{RE} in comparison to solving Least-Squares on the full data set, while r has to be proportional to n in order to get the same bound for C_{PE}. [1] refers to a theorem of another paper that contains a lower bound on C_{PE}, showing that the bound of the sampling-based sketching algorithm cannot be improved in the general case [8]. This theorem is stated below.

Theorem 3. *(Lower bound on Prediction Efficiency) For any Sketching matrix satisfying* $\mathbb{E}[\|S^T(SS^T)^{-1}S\|_2] \leq \eta\frac{r}{n}$*, any estimator based on (SX, SY) satisfies the following lower bound with probability* > 0.5*:*

$$C_{PE}(S) \geq \frac{n}{128\eta r}$$

While this theorem holds in general for a dataset (X ,Y), [1] considers a special case where the leverage scores of the data matrix X are highly skewed to only a small number of data points. First, let us define this special-case assumption on X.

Definition 5. *(K-heavy hitter leverage distribution) A sequence of leverage scores is a k-heavy hitter leverage distribution if for* $1 \leq i \leq k$, $\frac{C_1 \times p}{k} \leq \ell_i \leq \frac{C_2 \times p}{k}$ *and* $\sum_{i=k+1}^{n} \ell_i \leq \frac{3}{4}$*, where p is the number of features in data matrix X and* $C_1, C_2 > 0$ *are constants.*

Intuitively, this means that just k data points in X have a high contributing factor to the prediction variance and the rest of them are less significant and will not affect the regression very much.

Given the assumption of a k-heavy hitter distribution, [1] shows that Algorithm 2, *i.e.* sampling according to leverage scores without rescaling, gives a tighter bound on C_{PE} and C_{RE}. The theorem is stated below, and the interested reader can find a detailed proof in the appendix of their paper.

Theorem 4. *(Upper bound: Random Sampling without Rescaling) For* $r \geq c_1 p \log(c_2 p)$*, with* $\theta = 0$*, and assuming a k-heavy hitter leverage distribution, then with probability* ≥ 0.6*:*

$$C_{RE}(S_{NR}) \leq 1 + \frac{44C^4}{c^2}\frac{pk}{nr}$$
$$C_{PE}(S_{NR}) \leq \frac{44C^4}{c^2}\frac{k}{r}$$

Note that the bounds in Theorem 3 and 4 are stated with non-parameterized probabilities (0.5 and 0.6). The parameterized bounds with variable confidence have a rather complex form and are less intuitive. The authors of [1] thus instantiated the bounds with fixed confidence to improve readability. However, the bounds are still useful. For example, for Theorem 4 we can run Algorithm 2 t times, and the probability that the bound does not hold for any execution is less than $(1-0.6)^t$, which can be made arbitrarily small.

In conclusion, the Sampling-with-rescaling method (Algorithm 1) is a general-purpose sketching algorithm that achieves optimal bounds on the sketched approximation of the Least Squares problem, where C_{RE} scales as $\frac{p}{r}$ and C_{PE} scales as $\frac{n}{r}$. However, in the non-scaled Sampling of Algorithm 2, where we assume the leverage distribution is highly skewed to only k examples, a much tighter bound can be achieved for C_{PE} that scales as $\frac{k}{r}$ (the assumptions violate the constraints of Theorem 2, whence the lower bound is not applicable).

2 Best-Arm Identification in Linear Bandits

In this section we study Linear Bandits (LB), which is an extension of the Multi-Armed Bandit (MAB) problem. Common to both settings, the player pulls an arm and receives some (typically stochastic) reward. In the case of best-arm identification, the goal is to identify, with as few pulls as possible, the arm associated with the highest expected reward. In the classic MAB setting, the reward of each arm is independent of all other arms (*i.e.* pulling an arm does not provide any information about the rewards of other arms). However, in many real-world problems it is more interesting if the arms share a set of common attributes or features. The reward of each arm is then modeled as a function of its specific attribute values. Of course, this function is not known to us, and identifying the best arm requires us to estimate the function. Specifically, in Linear Bandits each arm is associated with a vector of features and the reward function is simply a weighted linear combination of these features. The set of (unknown) linear weights, called the parameter vector, remains constant and is shared for all arms. For example, if each arm represents a distinct type of car, a reasonable feature vector might encode the attributes of that car (e.g. top speed, acceleration, if it has four-wheel drive, etc.). If the observable reward of the bandit game is the time it takes for a car to travel from location A to B, the unknown parameter vector would encode the weighted relationship between each feature and the total travel time. This class of problems can further be augmented by considering generalized linear bandits (GLB) [9] which allow the rewards to be non-linear functions of the parameter vector and arm features, such as a logistic curve for binary rewards. Also note that we can think of the original MAB problem as a special case of the LB problem where each arm has a single unique feature equal to one and all other features corresponding to different arms are zero. The weights are then the expected reward received from each arm.

A natural application of linear bandits is planning a route where we have several possible paths and there is some unknown stochastic cost to each leg of the path, but we might only observe the total cost of the journey. Similarly, we might want to identify customers' preferences presented by a feature vector by suggesting products and then observing which ones they click on, similar to Amazon's Shop-by-look project (https://shopbylook.amazon.com/). Just as with the MAB problem we could minimize overall regret using an approach similar to UCB, which is called Optimism in the Face of Uncertainty (OFUL) for linear bandits [10]. However, we will focus on the best arm identification problem where our goal is to identify the arm with the highest expected reward within as few arm pulls as possible. Here we first introduce static allocation strategies for the problem (strategies that decide which sequence of arms to pull before starting the bandit game) and demonstrate how to relate this to G-optimality of linear experiment design. Then we show that a dynamic allocation, where the decision of which arm to pull changes adaptively based on previous arm pulls, can reduce the sample complexity of the problem. These strategies are summarized mainly from [11].

2.1 Setting

In the Linear Bandit (LB) problem, there are K arms to play, and each arm is associated with a d-dimensional feature vector $x \in \mathbb{X}$, where $\mathbb{X} \subset \mathbb{R}^d$ is the set of feature vectors corresponding to arms ($|\mathbb{X}| = K$). The feature vector x of an arm is fully observable (known). As the player plays an arm with feature vector x, a random reward $r(x)$ is received and observed. Reward $r(x)$ is generated according to the linear model $r(x) = \theta_*^T x + \epsilon$, where $\theta_* \in R^d$ is an unknown parameter vector and $\epsilon \sim N(0, \sigma^2)$ is zero-mean i.i.d noise. That is, the expected reward is a linear combination of the features x of that arm, weighted by the unobservable parameters of θ_*. In contrast to the classic Multi-Armed Bandit setting where the rewards of different arms are independent, here the parameter vector θ_* is shared between all arms $i \in [K]$. Because of the linear structure of the reward $r(x)$, by playing an arm the player can estimate the true value of θ_* and, indirectly, gain information about the expected reward of other arms (which is $\theta_*^T \tilde{x}$ for any other arm with feature vector $\tilde{x}$). This in turn allows us to estimate the arm with the highest expected reward. The goal is to identify the best arm with as few arm pulls as possible. The optimal arm is defined as:

$$x^* = arg \max_{x \in \mathbb{X}} \theta_*^T x \tag{1}$$

Of course, since θ_* is unknown, we are unable to solve this optimization problem directly. Instead, in the sections below we present strategies for estimating θ_* and choosing a sequence of arms to play so that x^*, the optimal arm, is found within as few arm pulls as possible. To aid in notation, we define $\prod(\theta) = \text{argmax}_{x \in \mathbb{X}} \theta^T x$ as the best-arm corresponding to an arbitrary parameter θ. We also define $\Delta(x, \tilde{x}) = (x - \tilde{x})^T \theta_*$, measuring the difference in reward between the feature vectors x_i and x_j of arms i and j.

Now, assume we have pulled arms for n time steps and denote x_t and r_t as the feature vector and reward of the arm pulled at step t. Let $\mathbf{x}_n = (x_1, ..., x_n)$ be the sequence (or *allocation*) of feature vectors corresponding to the pulled arms and let $\mathbf{r}_n = (r_1, ..., r_n)$ be the corresponding sequence of rewards. By treating the sequence $\mathbf{x}_n$ as input feature vectors and $\mathbf{r}_n$ as output response variables, we can apply Ordinary Least Squares (Lemma 1) to estimate the value of θ_*:

$$\hat{\theta}_n = A_{x_n}^{-1} b_{x_n} \tag{2}$$

Here $A_{x_n} = \sum_{t=1}^n x_t x_t^T \in R^{d \times d}$ is simply the Information matrix of the linear system ($X^T X$ if we were to stack the feature vectors x_i as rows in matrix X), $b_{x_n} = \sum_{t=1}^n x_t r_t \in R^d$ and $\hat{\theta}_n$ is the OLS estimate of θ_* obtained after n arm pulls. Now, through Azuma's inquality [12], the prediction error for this estimate can be bounded in high probability by:

$$\mathbb{P}(\forall n \in \mathbb{N}, \forall x \in \mathbb{X}, |x^T(\theta_* - \hat{\theta}_n)| \leq c||x||_{A_{\mathbf{x}_n}^{-1}} \sqrt{log(K^2/\delta)}) \geq 1 - \delta \tag{3}$$

In the above probability bound, K is the number of different arms of the linear bandit problem, c is a constant and n is the number of arms pulled (*i.e.* the number of data points used in the regression). $\delta \in (0, 1)$ is the user-selected confidence parameter, and $||x||_{A_{\mathbf{x}_n}^{-1}}$ is the induced vector norm in the matrix space $A_{\mathbf{x}_n}^{-1}$ (For a positive semidefinite matrix A and vector x, $||x||_A = \sqrt{x' A x}$). This bound on the estimation error of $\hat{\theta}_n$ will be readily used in the following sections for the sample complexity required to identify the optimal arm.

2.2 Oracle strategy

In this section we present a strategy from Soare et al. [11] for selecting which arms of the linear bandit to sample in order to identify the optimal arm (with high probability) in a bounded number of steps. In this strategy, the sequence of which arms to sample is chosen a priori (before starting the bandit game, and before observing any rewards). The strategy, however, depends on the existence of an "oracle" with access to the true parameter vector θ_* that we can query. As it turns out, the strategy developed here can instead of θ_* be used with parameter estimates $\hat{\theta_n}$ and still achieve a good upper bound on the sample complexity.

2.2.1 Stopping criterion

Let us define $C(x) = \cap_{\tilde{x} \in \mathbb{X}}\{\theta \in R^d, (x-\tilde{x})^T\theta \geq 0\}$ as the set of parameters θ for which the arm with feature vector x is optimal. It is easy to see that this definition is correct by observing the condition $(x-\tilde{x})^T\theta \geq 0$, which implies that given θ, feature vector x leads to higher reward than all other feature vectors $\tilde{x}$.

Now, suppose we had an oracle with access to $C(x^*)$, which is the set of all parameters θ where the optimal arm x^* (for θ^*) is still optimal ($\theta^T x^* \geq \theta^T x$ for $\forall x$). Also assume that we can build a confidence set $S^*(\mathbf{x}_n)$ of likely parameter choices θ around θ^* such that with probability $\geq 1-\delta$, for any possible sequence of arm pulls $\mathbf{x}_n$ it includes our estimate $\hat{\theta}_n$. We can use this confidence set together with $C(x^*)$ to determine whether we have identified the optimal arm x^*. Specifically, if the confidence set $S^*(\mathbf{x}_n)$ is completely contained within $C(x^*)$ (*i.e.* $S^*(\mathbf{x}_n) \subseteq C(x^*)$), we have successfully identified the optimal arm and can hence stop. The reason for this is that if all possible values of $\hat{\theta}_n$ (which is our confidence set $S^*(\mathbf{x}_n)$) lies completely within the set of parameters where x^* is optimal ($C(x^*)$), then clearly $\text{argmax}_{x \in \mathbb{X}} \hat{\theta}_n^T x$ will return the optimal arm x^* for any $\hat{\theta}_n \in S^*(\mathbf{x}_n)$. Figure 3 further illustrates the intuition behind this stopping condition.

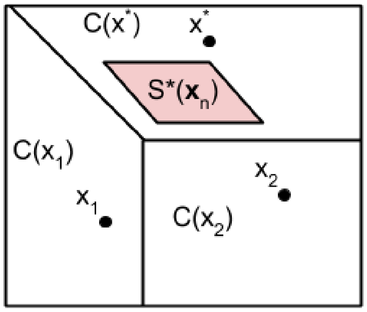

Figure 3: The three sets $C(x_1)$, $C(x_2)$ and $C(x^*)$ corresponding to the arms x_1, x_2, x^*, which partition the space of possible parameters θ. When the confidence set $S^*(\mathbf{x}_n)$ around θ_* (colored in pink) is completely contained in $C(x^*)$, there is no ambiguity regarding which arm is the optimal one, since all possible parameter estimates $\hat{\theta}_n$ in the confidence set $S^*(\mathbf{x}_n)$ agree it is x^*.

2.2.2 Arm selection strategy

From the previous section we found out that we want the confidence set $S^*(\mathbf{x}_n)$ to become contained within $C(x^*)$, as this implies we have identified the optimal arm (the *stopping condition*). Then, clearly, our strategy should be to shrink $S^*(\mathbf{x}_n)$ into $C(x^*)$ with as few arm pulls as possible. The condition $S^*(\mathbf{x}_n) \subseteq C(x^*)$ implies that for every possible θ in our confidence set $S^*(\mathbf{x}_n)$, the reward $\theta^T x$ should be smaller than $\theta^T x^*$ for all arms x. If we define $\mathcal{Y}^* = \{y = x^* - x, \forall x \in \mathbb{X}\}$ as the set of d-dimensional vectors pointing to the optimal arm x^* and $\Delta(y) = \Delta(x^*, x) = (x^* - x)^T\theta_*$ as the reward gap of x, the inequality can be written as:

$$\begin{aligned} \theta^T x &\leq \theta^T x^*, \forall x \in \mathbb{X}, \forall \theta \in S^*(\mathbf{x}_n) \\ 0 &\leq \theta^T(x^* - x) \\ \theta_*^T(x^* - x) &\leq \theta^T(x^* - x) + \theta_*^T(x^* - x) \\ \theta_*^T(x^* - x) - \theta^T(x^* - x) &\leq \theta_*^T(x^* - x) \\ y^T(\theta_* - \theta) &\leq \Delta(y), \forall y \in \mathcal{Y}, \forall \theta \in S^*(\mathbf{x}_n) \end{aligned}$$

Hence, if $y^T(\theta_* - \theta) \leq \Delta(y)$ for all y and θ, we have identified the optimal arm $x^* = \text{argmax}_{x \in \mathbb{X}} \theta^T x$ for any $\theta \in S^*(\mathbf{x}_n)$. We can take advantage of the upper bound on the estimation error defined in Equation 3 to construct a confidence set that fulfills this inequality with high probability. Specifically, define $S^*(\mathbf{x}_n)$ as:

$$S^*(\mathbf{x}_n) = \{\theta \in R^d, \forall y \in \mathcal{Y}^*, y^T(\theta^* - \theta) \leq c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{log(K^2/\delta)}\} \tag{4}$$

From Equation 3, with probability $\geq 1 - \delta$, the inequality $y^T(\theta_* - \theta) \leq c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{log(K^2/\delta)}$ holds. Hence, our stopping condition is met once the upper bound $c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{log(K^2/\delta)}$ on our confidence set is smaller than the RHS of the stopping condition ($\Delta(y)$):

$$c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{log(K^2/\delta)} \leq \Delta(y) \tag{5}$$

To make this condition hold as quickly as possible, the oracle strategy should simply select a sequence of arm pulls $\mathbf{x}_n$ that minimizes the ratio of the LHS and RHS. This can be simplified to:

$$\mathbf{x}_n^* = arg \min_{\mathbf{x}_n} \max_{y \in \mathcal{Y}^*} \frac{||y||_{A_{\mathbf{x}_n}^{-1}}}{\Delta(y)} \tag{6}$$

Having defined an oracle strategy, we want to find out the minimum number of samples (or arm pulls) needed to achieve the oracle stopping criterion defined previously. However, the discrete optimization problem in Equation 6 is hard to solve and can instead be lower bounded by a 'soft-allocation' formulation, defined as:

$$H_{LB} = \min_{\lambda \in D^k} \max_{y \in \mathcal{Y}^*} \frac{||y||^2_{\Lambda_\lambda^{-1}}}{\Delta^2(y)} \tag{7}$$

Here, $\lambda \in \mathbf{D}^k$ denotes the proportions of pulls to an arm x, $\mathbf{D}^k$ denotes the simplex $\mathbb{X}$ and $\Lambda_\lambda = \sum_{t=1}^n \lambda_{I_t} x_t x_t^T$ (I_t is the index of the arm pulled at time t). The sample complexity of the oracle strategy (denoted N^*) is

$$N^* = c^2 H_{LB} log(K^2/\delta)$$

where K is the number of arms, c is a constant and δ is the confidence level that the best arm will be identified. A more thorough derivation of this bound can be found in [11]. While it may currently seem unnecessary to have an oracle strategy and an oracle lower bound for sample complexity, we will soon see that this provides a good benchmark to compare the realistic algorithms against.

2.3 Static allocation strategies

While the oracle strategy provides a theoretical foundation for the complexity of best-arm identification in Linear Bandits, it is in practice not applicable as an algorithm for solving the identification problem. This is because the oracle strategy assumed we had access to the true (unknown) parameter vector θ_* as well as the set of direction vectors $y = x^* - x$ of $\mathcal{Y}^*$ which point toward the optimal arm x^*. Hence, we first need to re-define the stopping criteria (Equation 5) in terms of measurable quantities that we have access to.

In the static allocation setting, where our algorithm determines the most optimal sequence $\mathbf{x}_n$ of arms to pull a priori (before starting the bandit game), [11] defines the empirical stopping criteria as follows: Let $\hat{S}(\mathbf{x}_n)$ be a high-confidence set around the parameter estimate $\hat{\theta}_n$ of possible choices for parameter values where the true parameter vector $\theta_* \in \hat{S}(\mathbf{x}_n)$ with probability $\geq 1 - \delta$. Furthermore, we re-use the definition of $C(x)$ as the set of possible parameters θ for which the arm with feature vector x is optimal. Now, if there exists some arm $x \in \mathbb{X}$ such that $\hat{S}(\mathbf{x}_n) \subseteq C(x)$, the stopping condition is met, because all possible parameter choices in our confidence set $\hat{S}(\mathbf{x}_n)$ all agree that $x = x^*$ is the optimal arm. Since we defined $\hat{S}(\mathbf{x}_n)$ to contain the true parameter vector θ^* with probability at least $1 - \delta$, it implies that the probability of our predicted optimal arm $\text{argmax}_{x \in \mathbb{X}} \hat{\theta_n}^T x$ not being optimal is at most δ ($\mathbb{P}(\text{argmax}_{x \in \mathbb{X}} \hat{\theta_n}^T x \neq x^*) \leq \delta$).

Similar to how we re-wrote the stopping condition for the oracle strategy, let us formulate the stopping condition $\hat{S}(\mathbf{x}_n) \subseteq C(x)$ as an inequality. First define the empirical reward gap between two arms as $\hat{\Delta}_n(x,\tilde{x}) = (x-\tilde{x})^T\hat{\theta}_n$. The empirical stopping criteria can then be re-written as:

$$\exists x, \forall \tilde{x} \in \mathbb{X}, \forall \theta \in \hat{S}(\mathbf{x}_n),\ (x-\tilde{x})^T\theta \geq 0 \tag{8}$$

$$\Leftrightarrow \exists x, \forall \tilde{x} \in \mathbb{X}, \forall \theta \in \hat{S}(\mathbf{x}_n),\ (x-\tilde{x})^T(\hat{\theta}_n - \theta) \leq \hat{\Delta}_n(x,\tilde{x}) \tag{9}$$

Equation 8 intuitively captures the stopping condition by stating that if there exists some arm x such that, for all other arms $\tilde{x}$ and all possible choices of the parameter θ in our confidence set $\hat{S}(\mathbf{x}_n)$, x has the highest expected reward, then x must be the optimal arm and we can successfully stop. Equation 9 is merely a reformulation of Equation 8 that expresses the inequality in terms of the directions $(x-\tilde{x})^T(\hat{\theta}_n-\theta)$ (where $x-\tilde{x}$ is a direction vector) and the empirical reward gaps $\hat{\Delta}_n(x,\tilde{x})$. Based on this inequality, we can again take advantage of Azuma's inequality (Equation 3) to construct our confidence set $\hat{S}(\mathbf{x}_n)$ of possible choices for parameter θ:

$$\hat{S}(\mathbf{x}_n) = \{\theta \in R^d, \forall x \in \mathbb{X}, \forall \tilde{x} \in \mathbb{X}, (x-\tilde{x})^T(\hat{\theta}_n - \theta) \leq c||x-\tilde{x}||_{A_{\mathbf{x}_n}^{-1}}\sqrt{log(K^2/\delta)}\} \tag{10}$$

This confidence set includes with probability $\geq 1-\delta$ the true parameter vector θ^*. Again similar to the oracle strategy, we can now state that the stopping condition holds whenever $c||x-\tilde{x}||_{A_{\mathbf{x}_n}^{-1}}\sqrt{log(K^2/\delta)}$ (the upper bound on our confidence set $\hat{S}(\mathbf{x}_n)$) is lesser than the empirical reward gaps $\hat{\Delta}_n(x,\tilde{x})$ for some arm x and all other arms $\tilde{x}$ (the reward gap was the upper bound on the stopping condition in Equation 9). Written compactly, the empirical stopping condition becomes:

$$\exists x, \forall \tilde{x} \in \mathbb{X}, c||x-\tilde{x}||_{A_{\mathbf{x}_n}^{-1}}\sqrt{log(K^2/\delta)} \leq \hat{\Delta}_n(x,\tilde{x}) \tag{11}$$

2.3.1 G-allocation strategy

Recall the G-optimal design criterion defined for Linear experimental design in Section 1.2 of this script. For this criterion, we minimized the function $f_G(X) = \max\ \text{diag}(X(X^TX)^{-1}X^T)$, which is the maximum diagonal element of the Hat matrix $X(X^TX)^{-1}X^T$, where X is our data matrix of input feature vectors and the minimization is done over the possible rows to include in X. We motivated its usefulness by observing that the criterion in fact minimizes the worst possible prediction variance. Now, we can easily relate this quantity to our stopping condition of Equation 11, by remembering that matrix $A_{\mathbf{x}_n}$ is defined as $A_{\mathbf{x}_n} = \sum_{t=1}^n x_t x_t^T$ over the arms x_t played in the sequence $\mathbf{x}_n$. That is, if we were to stack our played feature vectors x_t in a matrix X, $A_{\mathbf{x}_n} = X^TX$ and $||x||_{A_{\mathbf{x}_n}^{-1}} = \text{diag}(x(X^TX)^{-1}x^T)$. Hence, we can formulate G-optimal design in our linear bandit notation as the sequence $\mathbf{x}_n$ of played arms which minimize the maximum value of $||x||_{A_{\mathbf{x}_n}^{-1}}$:

$$\mathbf{x}_n^G = arg \min_{\mathbf{x}_n} \max_{x \in \mathbb{X}} ||x||_{A_{\mathbf{x}_n}^{-1}} \tag{12}$$

Returning to the problem of best-arm identification for linear bandits, we make two observations. First, we note that the LHS of the stopping condition of Equation 11 depends on the term $||x-\tilde{x}||_{A_{\mathbf{x}_n}^{-1}}$. Second, we note that for any pair of arms (x_1, x_2), it is always true that $||x_1 - x_2||_{A_{\mathbf{x}_n}^{-1}} \leq 2\max_{x\in\mathbb{X}}||x||_{A_{\mathbf{x}_n}^{-1}}$. Hence, minimizing $\max_{x\in\mathbb{X}}||x||_{A_{\mathbf{x}_n}^{-1}}$ will help us achieve the stopping condition by minimizing an upper bound. Upon further inspection, this is identical to the G-optimality criterion and hence methods used to solve the G-optimal design problem can be directly used here. We call this strategy *G-allocation.*

Practical implementation of the G-allocation strategy

The G-allocation problem is a combinatorial optimization problem which we previously have argued is NP-hard. There are however approximation methods for achieving G-optimal design. [11] mentions two particular approaches, namely i) continuous relaxation strategy and ii) greedy incremental arm selection strategy.

The continuous relaxation problem is defined as:

$$\lambda^G = arg \min_{\lambda \in D^k} \max_{x \in \mathbb{X}} ||x||_{\Lambda_\lambda^{-1}} \tag{13}$$

Having obtained the optimal λ^G, which is a K-dimensional vector (K is the number of arms) of proportions of arm pulls, the sequence $\mathbf{x}_n$ can be approximated by (roughly speaking) rounding the proportion vector λ^G into a vector of integers denoting the number of times an arm should be pulled. For details of efficient rounding procedures, we refer the reader to [11]. The next approximation method is the greedy incremental arm selection strategy, which greedily chooses the next arm to include in the sequence of arms to play by locally maximizing the G-optimality criterion only for the next arm, given the previous history of played arms. The t-th iteration of this algorithm can be expressed as:

$$x_t = arg \min_{x \in \mathbb{X}} \max_{\tilde{x} \in \mathbb{X}} \tilde{x}^T (A_{x_{t-1}} + xx^T)^{-1} \tilde{x} \tag{14}$$

For both methods, the approximation is bounded by a factor $(1+\beta)$ from the optimal value, where β is a constant. This leads to the following probabilistic claim about the sample complexity for the G-allocation strategy (denoted N^G):

$$\mathbb{P}[N^G \leq \frac{16c^2 d(1+\beta) log(K^2/\delta)}{\Delta^2_{min}} \wedge \text{argmax}_{x \in \mathbb{X}} \big(\hat{\theta}^T_{N^G} x\big) = x^*] \geq 1-\delta \tag{15}$$

Here c is a constant, K is the number of arms, d is the feature vector dimensionality of the arms, $\hat{\theta}_{N^G}$ is the parameter vector estimate obtained through performing OLS on the arms (feature vector-reward pairs) sampled by the G-allocation strategy, and δ is the user-supplied confidence level that the best arm is identified. The left side of the intersection bound limits the sample complexity N^G, while the right side states that the predicted best arm $\text{argmax}_{x \in \mathbb{X}}(\hat{\theta}^T_{N^G} x)$ is indeed the optimal arm x^*.

2.4 Dynamic allocation strategies

Using a dynamic allocation strategy, that is, a strategy which adapts to the observed rewards online, we can significantly decrease the sample complexity to approach the oracle solution. The idea is to adapt the allocation strategy to discard suboptimal arms online. Now we only keep track of the potentially optimal arms $\hat{\mathbb{X}}(\mathbf{x}_n)$ (where $\mathbf{x}_n$ is the allocation sequence of arm pulls) and discard any others using the test below. Here we use the empirical reward gap $\hat{\Delta}_n(x, \tilde{x}) = (x - \tilde{x})^T \hat{\theta}_n$, where $\hat{\theta}_n$ is our estimate for θ^*.
Arm $x \in \mathbb{X}$ is dominated (*i.e.* it is with probability $\geq 1-\delta$ suboptimal compared to the other arms) when:

$$\exists \tilde{x} \in \mathbb{X} \; s.t. \; c||\tilde{x} - x||_{A^{-1}_{\mathbf{x}_n}} \sqrt{\log(K^2/\delta)} < \hat{\Delta}_n(\tilde{x}, x) \tag{16}$$

The condition states that when the upper confidence bound on the reward of arm x is strictly smaller than the reward gap to some other arm $\tilde{x}$, x is suboptimal with high probability. This bound however, which is derived from Equation 3 (see Proposition 1 in [11] for more details), is only valid when the sequence $\mathbf{x}_n$ of arms to pull is determined a priori (before pulling any arms). As soon as we start choosing (or discarding) arms adaptively based on previous rewards, the bound does no longer hold. Consequently, [11] derives the following adaptive upper bound on the prediction error with regularizer η and $\tilde{A}^\eta_{\mathbf{x}_n} = \eta I_d + A_{\mathbf{x}_n}$ (see Proposition 2 in [11] for more details). With probability $\geq 1-\delta$, the following bound holds:

$$|x^T \theta_* - x^T \hat{\theta}_n| \leq ||x||_{(A^\eta_{\mathbf{x}_n})^{-1}} \left(\sigma \sqrt{d \log \left(\frac{1 + nK^2/\eta}{\delta} \right)} + \eta^{1/2} ||\theta_*|| \right) \tag{17}$$

Here d is the feature vector dimensionality of the arms, η is a regularizing constant, σ^2 is the variance of the noise observed in the rewards, n is the number of arms pulled and K is the number of arms. Finally, note that this bound holds when x_t, the arm sampled at time t, depends on (x_1, r_1, ..., x_{t-1}, r_{t-1}). Comparing

Equation 17 to 3, with an adaptive strategy the sample complexity scales linearly with the dimensionality d. This would suggest there is actually no advantage in using an adaptive strategy. The solution is to split the algorithm into phases during which we do not change $\hat{\mathbb{X}}(x_n)$, the set of arms that we are still considering after having sampled the arms in the sequence $\mathbf{x}_n$. This way we can treat the strategy as quasi-static (the sequence $\mathbf{x}_n$ of arms to sample is chosen statically between phases). Since $\hat{\mathbb{X}}(x_n)$ is only changing between phases, it allows us to approach the oracle sample complexity of Equation 7. For the jth phase of length n_j (n_j arms are sampled during this phase), given the set of remaining non-dominated arms $\hat{\mathbb{X}}_j$ after phase $j-1$, the allocation implemented at phase j is:

$$\mathbf{x}^j_{n_j} = arg \min_{\mathbf{x}_{n_j}} \max_{x,\tilde{x} \in \hat{\mathbb{X}}_j} ||x - \tilde{x}||_{A^{-1}_{\mathbf{x}_{n_j}}} \tag{18}$$

The motivation for this allocation rule is that if we minimize the maximum norm $||x - \tilde{x}||_{A^{-1}_{\mathbf{x}_{n_j}}}$ between any pair of arms $x, \tilde{x}$, we are directly shrinking the upper bound in the stopping condition of Equation 17. At the end of the phase, we re-estimate the parameter vector $\hat{\theta_n}$ based on the sequence of sampled arms and use $\hat{\theta_n}$ together with Equation 16 to discard arms from $\hat{\mathbb{X}}_j$.

Now we must choose a good phase length because if the phase is too short, the estimate of $\hat{\theta_n}$ will not be good enough to discard an arm. On the other hand, if the phases are too long, we do not take advantage of the adaptivity. We therefore choose a phase length n_j such that the uncertainty of all the directions of interest decrease by a factor $\alpha \in (0, 1)$. Using the objective $\rho^j(\lambda) = \max_{x,\tilde{x} \in \hat{\mathbb{X}}_j} ||x - \tilde{x}||^2_{\Lambda^{-1}_{\lambda}}$, [11] defines:

$$n_j = \min \left\{ n \in N : \frac{\rho^j(\lambda_{\mathbf{x}^j_n})}{n} \leq \frac{\alpha \rho^{j-1}(\lambda^{j-1})}{n_{j-1}} \right\} \tag{19}$$

where $\mathbf{x}^j_n$ is the allocation defined in Equation 18, λ_{j-1} is the allocation performed at phase $j-1$ (i.e. $\mathbf{x}^{j-1}_{n_{j-1}}$), n_{j-1} is the number of arms sampled at phase $j-1$ and N is the maximum number of arm pulls we consider.

At the beginning of the bandit game, $\hat{\theta_n}$ is very uncertain and as a result, a large portion of the sample complexity will come from discarding sub-optimal arms. The remaining component of the complexity is simply the same as for the oracle strategy. Performing the adaptive allocation strategy with a β-approximate method (as defined in Section 2.3.1), the resulting overall sample complexity (denoted N) would be as follows:

$$\mathbb{P}\left[\{ N \leq \frac{(1+\beta) \max \left\{ M^*, \frac{16}{\alpha} N^* \right\}}{\log(1/\alpha)} \log \left(\frac{c\sqrt{\log(K^2/\delta)}}{\Delta_{min}} \right) \} \wedge \{ \text{argmax}_{x \in \hat{\mathbb{X}_n}} \hat{\theta}^T_n x = x^* \} \right] \geq 1 - \delta \tag{20}$$

Here N^* is the oracle complexity, M^* is a term originating from the arm elimination process that depends on how distinguishable two arms are (in terms of their feature vectors), c, α, β are constants, Δ_{min} is the minimum reward gap between two arms, K is the number of arms and δ is the user-supplied confidence level that the best arm is identified. The left side of the intersection bound limits the sample complexity N, while the right side ensures that the predicted optimal arm $\text{argmax}_{x \in \hat{\mathbb{X}_n}} \hat{\theta}^T_n x$ is indeed optimal. Notice that the sample complexity N for the adaptive allocation strategy does not depend on the feature vector dimensionality d, which is a very powerful result as it allows us to handle arbitrarily high-dimensional Linear Bandit problems with little to none added complexity.

In conclusion, we showed that in the static setting where we choose which arms to pull before starting the bandit game, the best-arm identification problem in Linear bandits can be reduced to G-optimal design. However, with such a strategy we incur a sample complexity that scales linearly with d, the feature vector dimensionality. Instead, we should use an adaptive allocation strategy by iteratively discarding arms based on previously observed rewards. With such a strategy, we showed that the sample complexity is independent of the dimensionality of the feature vectors.

References

[1] Garvesh Raskutti and Michael Mahoney. A statistical perspective on randomized sketching for ordinary least-squares, 2014.

[2] Press H. William et. al. Numerical recipes in c: The art of scientific computing, 1992.

[3] Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal design of experiments via regret minimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 126–135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[5] Friedrich Pukelsheim. *Optimal Design of Experiments (Classics in Applied Mathematics) (Classics in Applied Mathematics, 50)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[6] Michael W. Mahoney. Randomized algorithms for matrices and data, 2011.

[7] Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Sarl. Faster least squares approximation. *Numer. Math.*, 117(2):219–249, February 2011.

[8] Mert Pilanci and Martin J. Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *J. Mach. Learn. Res.*, 17(1):1842–1879, January 2016.

[9] Sarah Filippi, Olivier Cappé, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 586–594, USA, 2010. Curran Associates Inc.

[10] Yasin Abbasi-Yadkori, Dvid Pl, and Csaba Szepesvri. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320, 2011.

[11] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, pages 828–836, 2014.

[12] Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.