CS 15-859: Algorithms for Big Data	Fall 2022
Lecture ?? : Notes on Online Lewis Weights Sampling	
Prof. David Woodruff	*Scribe: Runze Wang*

1 Recap

1.1 Online Leverage Scores

Recall that in Problem Set 2, Problem 1 we studied online leverage scores.

Suppose we have an $n \times d$ matrix A with $n \geq d$ and with rank d. Let a_i denote the i-th row of A, and let A_i denotes the submatrix of A formed by the first i rows of A. The (offline) leverage score of row i is given by

$$\tau_i(\boldsymbol{A}) = \ell_i = a_i^T (A^T A)^{-1} a_i.$$

The i-th online (λ-ridge) leverage score [1,3] is defined to be

$$\ell_i^{\mathtt{OL}} = \min(a_i^T (A_{i-1}^T A_{i-1} + \lambda I)^{-1} a_i, 1)$$

In Problem set 2, we have shown that we can bound how much ℓ_i blows up. A similar bound is presented in [3]:

$$\sum_{i=1}^{n} \ell_i = O(d \log(1 + \|A\|_2^2 / \lambda)).$$

1.2 ℓ_p Norm Subspace Embedding

Problem 1 of Problem Set 1 shows how one can construct ℓ_p norm subspace embedding via ℓ_2 norm subspace embeddings when p is an even integer. We know that there are many choices of sketching matrices to construct ℓ_2 norm subspace embeddings. Specifically, using the notations in the problem, given vector $\boldsymbol{y} \in \mathbb{R}^d$, let $\boldsymbol{y}^{\otimes p} \in \mathbb{R}^{d^p}$ be the vector with $(i_1, \ldots, i_p)$-th entry equal to $\boldsymbol{y}_{i_1} \cdot \boldsymbol{y}_{i_2} \cdots \boldsymbol{y}_{i_p}$ where $i_\ell \in \{1, 2, \ldots, d\}$ for $\ell \in \{1, \ldots, p\}$. It follows that if we set $M = \begin{bmatrix} \boldsymbol{a}_1^{\otimes p/2} \\ \cdots \\ \boldsymbol{a}_n^{\otimes p/2} \end{bmatrix}$, we get

$$\|\boldsymbol{A}\boldsymbol{x}\|_p^p = \left\|\boldsymbol{M}\boldsymbol{x}^{\otimes p/2}\right\|_2^2.$$

Thus, if $\boldsymbol{S}$ is a random sketching matrix that offers a ℓ_2 norm subspace embedding for the column space of $\boldsymbol{M}$, we automatically obtain an ℓ_p norm subspace embedding for the column space of $\boldsymbol{A}$:

$$\left\|\boldsymbol{S}\boldsymbol{M}\boldsymbol{x}^{\otimes p/2}\right\|_2^2 \approx (1 \pm \epsilon) \left\|\boldsymbol{M}\boldsymbol{x}^{\otimes p/2}\right\| = (1 \pm \epsilon) \|\boldsymbol{A}\boldsymbol{x}\|_p^p.$$

While this approach might require $\boldsymbol{S}$ to have $O(d^{p/2}\epsilon^{-2})$ rows, the factor $d^{p/2}$ cannot be further optimized. One problem with this approach though is we are assuming we are working with even integers, not general positive p.

In this scribe notes, we aim at generaling both results. In particular, we will show that ℓ_p norm Lewis weights allow us to construct ℓ_p norm subspace embeddings, and we can generalize offline Lewis weights to an online setting. The seminal work of Cohen and Peng [4] prove that given a matrix $\boldsymbol{A}$, we can approximate ℓ_p norm Lewis weights for rows of $\boldsymbol{A}$ in $\tilde{O}(\mathrm{nnz}\boldsymbol{A} + d^{\Theta(p)})$ for arbitrary $p \in (0, +\infty)$. Here, $\tilde{O}$ hides a polynomial factor in p. This runtime is not ideal when p is large. The results of [6] show that we can compute ϵ-approximate ℓ_p Lewis weights in $\tilde{O}(poly(\epsilon)(\mathrm{nnz}\boldsymbol{A} + d^{\omega}))$ time, providing bounds when p is large. As we should see, the results of [8] not only showed it is possible to generalize ℓ_p norm Lewis weights to an online setting: the results also proved that we can obtain stronger bounds on the samples required to achieve ℓ_p norm subspace embeddings.

2 Offline Lewis Weights

Before diving into online lewis weights, we should study the definition of offline Lewis weights.

Definition ((Offline) Lewis Weights, [4])**.** Given an $n \times d$ matrix $\boldsymbol{A}$ and norm p, the ℓ_p Lewis weights $\overline{\boldsymbol{w}}$ are the unique weights such that for each row i, we have

$$\boldsymbol{a}_i^\top (\boldsymbol{A}^\top \overline{\boldsymbol{W}}^{1-2/p} \boldsymbol{A})^{-1} \boldsymbol{a}_i = \overline{\boldsymbol{w}}_i^{2/p} \tag{1}$$

or alternatively,

$$\overline{\boldsymbol{w}}_i = \tau_i(\boldsymbol{W}^{1/2-1/p}\boldsymbol{A}). \tag{2}$$

Notice if we plug in $p = 2$, we immediately get the expression for leverage scores. Thus, we know that ℓ_2 norm Lewis weights are exactly the leverage scores. A special cass is ℓ_1 norm Lewis weights. As we shall show next, ℓ_1 norm Lewis weights give us nicer ℓ_1 norm subspace embedding guarantees than the result we have seen in class using a dense matrix of scaled random Cauchy variables.

Here is a way to understand the intuition behind such a definition, inspired from the approach in [4]. First, if we look at Equation (2), we see it essentially says: let us assign a weight $\boldsymbol{w}_i$ for each row, such that after scaling $\boldsymbol{a}_i$ by $\boldsymbol{w}_i^{1/2-1/p}$, the leverage score of scaled i-th row is exactly $\boldsymbol{w}_i$ we started up with.

Now let us think about the simplest case that motivates our study, namely we have k copies of duplicate rows $\boldsymbol{a}_i$. We hope after reweighting, those duplicate copies should represent a single row of $\boldsymbol{a}_i$. Notice that if we want to rescale the k copies of $\boldsymbol{a}_i$ so that the final p-norm is the same, we should multiply $k^{-1/p}$ to each of the duplicate rows.

Now what if we do not exactly get k duplicate copies of $\boldsymbol{a}_i$? In this case, we still (by [4]) consider there are actually $\boldsymbol{w}_i$ copies of $\boldsymbol{a}_i$, so we should assume each row should first be scaled as $\boldsymbol{w}_i^{-1/p}\boldsymbol{a}_i$. Now if switching back to ℓ_2, original row i of $\boldsymbol{A}$ will correspond to $\boldsymbol{w}_i$ rows of $\boldsymbol{w}_i^{-1/p}\boldsymbol{a}_i$, but the effect of these rows (to ℓ_2 norm) is now equal to one row of $\boldsymbol{w}_i^{1/2-1/p}\boldsymbol{a}_i$. Now how to choose the weights? We say in this case, $\boldsymbol{w}_i^{1/2-1/p}\boldsymbol{a}_i$ should have a leverage score (which is a ℓ_2 norm thing) equal to $\boldsymbol{w}_i$. This gives us Equation 2.

Unfortunately, equation 1 or 2 are not convenient since when $p \neq 2$, the weights appear on both sides of the equation. Fortunately, this issue can be resolved by rephrasing the definition of Lewis weights in terms of convex optimization problems. Some possible reformulations are given in [4] and [7].

3 Properties of Lewis Weights

Here we will present some properties and non-properties of Lewis weights. Notice when we are dealing with arbitrary ℓ_p norm, it is usually helpful to distinguish between the cases of $p \in (0, 1)$, $p = 1$, $p \in (1, 2)$, $p = 2$, and $p > 2$. We should see that ℓ_p norm Lewis weights can exhibit some counterintuitive properties.

3.1 Monotonicity

Leverage scores exhibit "monotonicity", which means when we add more rows to the matrix, the leverage score of given rows won't increase. In other words, the relative importance of each row can only decrease once we see more rows of the matrix, which is naturally the case shall we want to repeat previously seen rows. Unfortunately, the monotonicity intution is only true for $p \leq 2$ [4], yet we can bound how much monotonicity is violated:

Lemma 3.1 (Lemma 5.6, [4])**.** *For all $p > 2$, if $\boldsymbol{A}'$ is $\boldsymbol{A}$ with any number of extra rows added,*

$$\boldsymbol{A}'^T \overline{\boldsymbol{W}}^{1/2-p} \boldsymbol{A}' \succeq d^{2/p-1} \boldsymbol{A}^T \overline{\boldsymbol{W}}^{1-2/p} \boldsymbol{A}$$

and in particular now row in $\boldsymbol{A}$ has its weights raised by no more than $d^{p/2-1}$.

This result can be a barrier in establishing bounds on online Lewis weights, since for online Lewis weights, we see the matrix $\boldsymbol{A}$ row by row, but adding later rows mean the weights of previous rows might increase.

3.2 Subspace Embedding Guarantees

Suppose we are sampling the rows of $\boldsymbol{A}$ with probability proportional to their ℓ_p norm Lewis weights (and appropriately scale the rows), we may obtain ℓ_p norm subspace embeddings for the column space of $\boldsymbol{A}$. In other words, let $\boldsymbol{S}$ be the sampling matrix, we have

$$\|\boldsymbol{S}\boldsymbol{A}\boldsymbol{x}\|_p = (1 \pm \epsilon) \|\boldsymbol{A}\boldsymbol{x}\|_p$$

In [4], it has been shown that to achieve a constant success probability, we need $O(\epsilon^{-2} d \log d)$ rows when $p = 1$ and $O(\epsilon^{-2} d \log(d/\epsilon) \log(\log d/\epsilon)^2)$ when $p \in (1, 2)$. The results in [2] shows when $p > 2$, the sample bound is $O(\epsilon^{-5} d^{p/2} \log d \log(1/\epsilon))$ for a failure probability of $O(poly(d^{-1}))$. One might wonder if $p > 2$, whether we can reduce the exponents over ϵ^{-1}. The current exponent of 5 is prohibitive should we need to achieve subspace embeddings with higher accuracy. The results in [8] shows that $\tilde{O}(\epsilon^{-2} d^{p/2})$ is actually possible where $\tilde{O}$ hids a polylog factor in δ^{-1}, n, and d:

Theorem 3.2 (Theorem 1.3, [8])**.** *Let $p > 2$ and $\boldsymbol{A} \in \mathbb{R}^{n\times d}$. Let $\delta \in (0,1)$ be a failure rate parameter and let $\epsilon \in (0,1)$ be an accuracy parameter. Let $\boldsymbol{w} \in \mathbb{R}^n$ be one-sided ℓ_p Lewis weights (i.e. overestimates of actual Lewis weights) with $\|\boldsymbol{w}\|_1 \leq O(d)$, which can be computed in $\tilde{O}(\mathrm{nnz}\boldsymbol{A} + d^\omega)$ time ([5, 6]). Let*

$$\alpha = O(d^{p/2-1}\epsilon^{-2}((\log d)^2(\log n) + \log\frac{1}{\delta})$$

be an oversampling parameter. Suppose that weights $\boldsymbol{s} \in \mathbb{R}^n$ are sampled by independently setting $\boldsymbol{s}_i = 1/\boldsymbol{p}_i^{1/p}$ with probability $\boldsymbol{p}_i := \min\{\alpha\boldsymbol{w}_i, 1\}$ and $\boldsymbol{s}_i = 0$ otherwise. Let $\boldsymbol{S} = diag(\boldsymbol{s})$. Then, with probability at least $1-\delta$, for all $\boldsymbol{x} \in \mathbb{R}^d$,

$$\|\boldsymbol{SA}\|_p = (1 \pm \epsilon)\|\boldsymbol{Ax}\|_p$$

and the sample complexity of $\boldsymbol{S}$ (i.e. the number of rows in $\boldsymbol{S}$, or the number of rows of $\boldsymbol{A}$ we are sampling) is at most

$$r = O(d^{p/2}\epsilon^{-2}((\log d)^2(\log n) + \log\frac{1}{\delta})).$$

3.3 Approximating Lewis Weights

Since one challenge of computing ℓ_p Lewis weights is the weights $\boldsymbol{w}$ appear on both sides of equation 1, we need to solve for an equation. Nevertheless, for these applications, one starting point is always solving the weights iteratively: we start with a possible candidate $\boldsymbol{w}$ of weights (for example, initialize $\boldsymbol{w} = \mathbf{1}$ at the start of the iteration scheme), and iterate using the equation 1. The results of [4] (Lemma 2.4) show this only work for $p \in (0,4)$. When $p \geq 4$, unfortunately, the iterative scheme fails since we are no longer dealing with a contracting map. Nevertheless, when $p \geq 2$, it is possible to use the convex optimization formulation of Lewis weights to achieve ϵ-approximate Lewis weights in polynomial time:

Theorem 3.3 (Theorem 4.4, [4])**.** *There exists a function $f(p,\epsilon)$ and a constant C such that for any $\boldsymbol{A}$, $p \geq 2$, $(1+\epsilon)$-approximate Lewis weights can be computed in time $O(f(p,\epsilon)n\log n\log\log nd^C)$.*

In this section, we have seen how Lewis weights can be applied to construct subspace embeddings for general ℓ_p norms. In particular, the guarantees for $p = 1$ is stronger than the guarantees we have seen in class using random Cauchy variables for sketching.

Finally, if we don't have exact Lewis weights but instead overestimates of Lewis weights, then have the following inequality for upper bounding ℓ_p norms of elements in the column space of $\boldsymbol{x}$ [5]. Here, by the definitions in [5], if the ℓ_p Lewis weights of $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ are $\boldsymbol{w}$, and $\boldsymbol{v}$ is a vector satisfying $d \leq \|\boldsymbol{v}\|_1 \leq O(1)d$ and $\boldsymbol{v}_i \geq \boldsymbol{w}_i$ for all i, then $\boldsymbol{v}$ is a Lewis weight overestimate.

Lemma 3.4 (Lemma 2.6, [5])**.** *For a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ and ℓ_p Lewis weight overestimates $\boldsymbol{w} \in \mathbb{R}^n_{>0}$, we have*

$$\|\boldsymbol{Ax}\|_p \leq \left\|\boldsymbol{W}^{1/2-1/p}\boldsymbol{Ax}\right\|_2$$

4 Online Lewis Weights

First, if we remove the ridge term in Equation 1.1, we may define online leverage scores to be (following the notations in [8])

$$\tau_i^{\texttt{OL}}(\boldsymbol{A}) := \begin{cases} \min\{\boldsymbol{a}_i^\top(\boldsymbol{A}_{i-1}^\top \boldsymbol{A}_{i-1})^- \boldsymbol{a}_i, 1\} & \boldsymbol{a}_i \in rowspan(\boldsymbol{A}_{i-1}) \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

Following the definition of Lewis weights in Equation 1, to construct online ℓ_p norm Lewis weights, it is natural to conjecture the following approach: we can simply insert term $\boldsymbol{W}^{p,\texttt{OL}}(\boldsymbol{A}_{i-1})^{1-2/p}$ where $\boldsymbol{W}^{p,\texttt{OL}}(\boldsymbol{A}_j)$ is an $j \times j$ diagonal matrices whose diagonal entries correspond to the online Lewis weights we have seen thus far (i.e the entry on the i-th row and column of $\boldsymbol{W}^{p,\texttt{OL}}(\boldsymbol{A}_j)$ is $\boldsymbol{w}_i^{p,\texttt{OL}}(\boldsymbol{A})$). Indeed, this is the approach taken in [8]:

$$\boldsymbol{w}_i^{p,\texttt{OL}}(A) := \begin{cases} \min\{[\boldsymbol{a}_i^\top(\boldsymbol{A}_{i-1}^\top(\boldsymbol{W}^{p,\texttt{OL}}(\boldsymbol{A})_{i-1})^{1-2/p}\boldsymbol{A}_{i-1})^- \boldsymbol{a}_i]^{p/2}, 1\} & \text{If } \boldsymbol{a}_i \in rowspan(\boldsymbol{A}_{i-1}) \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

First, we would like to see what is the blowup on the sum of online Lewis weights. We need to introduce the notion of online condition number [8]:

Definition (Online Condition Number, [8])**.** Let $\boldsymbol{A} \in \mathbb{R}^{n\times d}$. The online condition number of $\boldsymbol{A}$ is defined to be

$$\kappa^{\texttt{OL}}(\boldsymbol{A}) := \|\boldsymbol{A}\|_2 \max_{1\le i\le n} \left\|\boldsymbol{A}_i^-\right\|_2 .$$

The results of [8] prove that just like the case for leverage scores, the blowup arising from online Lewis weights is bounded by $\log(n\kappa^{\texttt{OL}})$:

Lemma 4.1 (Lemma 3.7, [8])**.** *Let $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ and $0 < p < \infty$. Then*

$$\sum_{i=1}^n \boldsymbol{w}_i^{p,OL}(\boldsymbol{A}) \le O(d\log(n\kappa^{OL}(\boldsymbol{A})))$$

As a result, we may use online Lewis weights to construct online ℓ_p norm subspace embeddings under the online coreset model, which allows us to *irrevocably* store a small subset of rows of the original matrix $\boldsymbol{A}$. Here, "online subspace embedding" means we are able to use matrices $\boldsymbol{S}_i$ to construct subspace embeddings for the column space of $\boldsymbol{A}_i$ for all i. The formal statement can be found in Theorem 3.8 of [8]:

Theorem 4.2 (Guarantees of online subspace embedding, Theorem 3.8, [8])**.** *Let $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ and $p = (0,\infty)$. Let δ be a failure rate parameter and let $\epsilon \in (0,1)$ be an accuracy parameter. Then there is an online coreset algorithm $\mathcal{A}$ such that, with probability at least $1-\delta$, $\mathcal{A}$ outputs a weighted subset of m rows with sampling matrix $\boldsymbol{S}$ such that*

$$\|\boldsymbol{S}_i\boldsymbol{A}_i\boldsymbol{x}\|_p^p = (1\pm\epsilon)\|\boldsymbol{A}_i\boldsymbol{x}\|_p^p . \tag{5}$$

where the sampling bound m is

$$O(d\epsilon^{-2}\log(n\kappa^{OL})poly(\log(n/\delta)))$$

if $p \in (0,2)$ *and*

$$O(d^{p/2}\epsilon^{-2}(\log(n\kappa^{\mathtt{OL}}))^{p/2+1} poly(\log(n/\delta)))$$

if $p \geq 2$ [1].

5 Proof of Online Lewis Weight Sampling

The proof of the blowup resulting from online Lewis weights in [8] is surprisingly simple. In fact, the proof itself reduces bounding sum of ℓ_p online Lewis weights to the bound on the sum of online leverage scores (as we have seen in Problem Set 2 or in [3]). The core observation is to realize that in equation 2, the Lewis weights are expressed as leverage score of a matrix resulted from scaling the i-th row of $\boldsymbol{A}$ by $\boldsymbol{w}_i^{1/2-1/p}$. The first step of proving Lemma 3.7 in [8] (i.e. the bound on the sum of online Lewis weights) is finding a bound in terms of $\kappa^{p,\mathtt{OL}}(\boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A})^{1/2-1/p}\boldsymbol{A})$:

Lemma 5.1. *Let* $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ *and* $p \in (0,+\infty)$*, then*

$$\sum_{i=1}^{n} \boldsymbol{w}_i^{p,\mathtt{OL}}(\boldsymbol{A}) \leq O(d \log \kappa^{\mathtt{OL}}(\boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A})^{1/2-1/p}\boldsymbol{A})) \tag{6}$$

Proof. First, let us think about the case where $\boldsymbol{w}_i^{p,\mathtt{OL}}(\boldsymbol{A}) < 1$. In this case, the only possibility is $\boldsymbol{a}_i$ is in the row span of $\boldsymbol{A}_{i-1}$, and in this case, after rearranging the term

$$\boldsymbol{w}_i^{p,\mathtt{OL}}(\boldsymbol{A}) = (\boldsymbol{w}_i^{p,\mathtt{OL}}(\boldsymbol{A})^{1/2-1/p}\boldsymbol{a}_i)^\top(\boldsymbol{A}_{i-1}^\top \boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A})_{i-1}^{1-2/p}\boldsymbol{A}_{i-1})^-(\boldsymbol{w}_i^{p,\mathtt{OL}}(\boldsymbol{A})^{1/2-1/p}\boldsymbol{a}_i)$$

Note here the transformation we are doing is quite similar to the definitions of Lewis weights in Equation 2.

Now notice that this essentially says $\boldsymbol{w}_i^{p,\mathtt{OL}}(\boldsymbol{A})$ is equal to the online leverage score for $\boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A})^{1/2-1/p}\boldsymbol{A}$, just like the offline ℓ_p Lewis weights $\boldsymbol{w}_i$ is equal to the offline leverage score of $\boldsymbol{W}^{1/2-1/p}\boldsymbol{A}$. Therefore, by plugging in the bound on online leverage scores in [3], we obtain the bound

$$\sum_{i=1}^{n} \boldsymbol{w}_i^{p,\mathtt{OL}}(\boldsymbol{A}) \leq O(d \log \kappa^{\mathtt{OL}}(\boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A})^{1/2-1/p}\boldsymbol{A}))$$

■

Now we have two cases for bounding $\kappa^{\mathtt{OL}}(\boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A})^{1/2-1/p}\boldsymbol{A})$. When $p \in (0,2)$, ℓ_p Lewis weights is monotonic, and thus online Lewis weights are actually greater than or equal to offline ℓ_p Lewis weights. Now we have the following chain of inequality:

$$\begin{aligned}
&\|\boldsymbol{A}_i x\|_2 \\
&\leq \left\|\boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A}_i)^{1/2-1/p}\boldsymbol{A}_i\boldsymbol{x}\right\|_2 \\
&\leq \left\|\boldsymbol{W}^{p}(\boldsymbol{A}_i)^{1/2-1/p}\boldsymbol{A}_i\boldsymbol{x}\right\|_2 \\
&\leq d^{|1/2-1/p|}\|\boldsymbol{A}_i\boldsymbol{x}\|_p \\
&\leq (nd)^{|1/2-1/p|}\|\boldsymbol{A}_i x\|_2 .
\end{aligned}$$

[1] I have rewritten the original bounds, which are given in terms of $\log d, \log n$ and $\log \frac{1}{\delta}$, using $poly(\log(n/\delta))$

Here, the second line uses the fact that if $p \in (0,2)$, $1/2 - 1/p < 0$, and the online Lewis weights are in the range $[0,1]$. The third line uses the fact that the offline ℓ_p Lewis weights are at most the online ℓ_p Lewis weights. The fourth line uses the fact that Lewis weights sum up to at most d (since we know that Lewis weights correspond to leverage scores of a scaled version of $\boldsymbol{A}$ and we may apply properties of leverage scores). And therefore,

$$\kappa^{\mathtt{OL}}\boldsymbol{A} = poly(n)\kappa^{\mathtt{OL}}(\boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A})^{1/2-1/p}\boldsymbol{A})$$

When $p > 2$, we no longer have monotonicity, but still we can use a variety of the property of Lewis weight overestimates given in [5]. In particular, we have the following chain of inequality:

$$\|\boldsymbol{A}_i\boldsymbol{x}\|_p \leq \left\|\boldsymbol{W}^{p,\mathtt{OL}}(\boldsymbol{A}_i)^{1/2-1/p}\boldsymbol{A}_i\boldsymbol{x}\right\|_2 \leq \|\boldsymbol{A}_i\boldsymbol{x}\|_2 .$$

Here, the first inequality is exactly the property about ℓ_p Lewis weight overestimates in [5], and the second inequality comes from the fact that in this case $1/2 - 1/p > 0$.

Thus, we can bound

$$\sum_{i=1}^{n} \boldsymbol{w}_i^{p,\mathtt{OL}}(\boldsymbol{A}) \leq O(d)\log(n\kappa^{\mathtt{OL}}(\boldsymbol{A})).$$

6 Pushing Towards Infinity

Obviously, at this point we are seeing that ℓ_p Lewis weights are not extremely helpful if p approaches infinity, given the sheer number of samples required. Nevertheless, we can still ask the following question: what is the limit when $p \to \infty$?

It turns out that this gives us the relationship between Lewis weights and John Ellipsoids.

Definition (Definition 5.4.2, [6])**.** For any matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$, we define the ℓ_p Lewis Ellipsoid of $\boldsymbol{A}$ by $E = \{\boldsymbol{x} \in \mathbb{R}^d | \boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{W}^{1-2/p}\boldsymbol{A}\boldsymbol{x} \leq 1\}$ where $\boldsymbol{w}$ is the ℓ_p Lewis weight of $\boldsymbol{A}$. For any convex set K, we define the John Ellipsoid of K be the maximum volume ellipsoid contained in K.

Surprisingly, the ℓ_∞ Lewis Ellipsoid of $\boldsymbol{A}$ corresponds to the John ellipsoid of $\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty \leq 1\}$ [6], and the reformulations of ℓ_p norm Lewis weight in terms of convex optimization problems [4, 6] is based on this intuition: we are trying to find an ellipsoid of maximum volume such that the sum of $\|\boldsymbol{A}\boldsymbol{x}\|_p^p$ inside the ellipsoid is bounded, and thus we are seeking a somewhat "softer" version of a John ellipsoid.

References

[1] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL: `https://proceedings.neurips.cc/paper/2015/file/f3f27a324736617f20abbf2ffd806f6d-Paper.pdf`.

[2] Jean Bourgain, Joram Lindenstrauss, and Vitali Milman. Approximation of zonoids by zonotopes. *Acta mathematica*, 162:73–141, 1989.

[3] Michael B Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. *Theory of Computing*, 16(APPROX-RANDOM 2016 Special Issue), 2020.

[4] Michael B. Cohen and Richard Peng. Lp row sampling by lewis weights. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC 15, pages 183–192, New York, NY, USA, 2015. Association for Computing Machinery. `doi:10.1145/2746539.2746567`.

[5] Arun Jambulapati, Yang P. Liu, and Aaron Sidford. Improved iteration complexities for overconstrained p-norm regression. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 529â542, New York, NY, USA, 2022. Association for Computing Machinery. `doi:10.1145/3519935.3519971`.

[6] Yin Tat Lee. *Faster Algorithms for Convex and Combinatorial Optimization*. PhD thesis, Massachusetts Institute of Technology, 2016.

[7] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in o (vrank) iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433. IEEE, 2014.

[8] David P Woodruff and Taisuke Yasuda. Online lewis weight sampling. *arXiv preprint arXiv:2207.08268*, 2022.