CS 514: Advanced Algorithms II – Sublinear Algorithms **Rutgers: Spring 2020**

Lecture 9

March 24, 2020

Instructor: Sepehr Assadi *Scribe: Chengyuan Deng and Runhui Wang*

1 Streaming Linear Regression

We move forward to exploring approaches of modeling streaming data in alien systems or algorithms. In this lecture, we extend the application of linear regression in a streaming model.

Recap of Linear Regression

As one of the baseline machine learning algorithms, Linear Regression models the linear relationship between feature variables and one target variable based on a data set of observations. Ideally, for a feature set $\boldsymbol{A}$ and label set $\boldsymbol{b}$ where $\boldsymbol{A}, \boldsymbol{b} \subseteq \mathbb{R}^n$, there exists a linear function f *s.t.* $\forall(a_i, b_i)$ with $a_i \in \boldsymbol{A}$ and $b_i \in \boldsymbol{b}$, $f(a_i) = b_i$.

In most of real-world scenarios, due to inevitable noise and bias in data collections, it would be almost hopeless to retrieve the *de facto* linear relationship between the features and label. However, it is possible to learn linear function $f : \hat{\boldsymbol{b}} = \boldsymbol{A} \cdot \boldsymbol{w}$ that minimize some distance measurement. Usually, we optimize a learning objective $\boldsymbol{L}(w)$ by minimizing total loss across all training samples based on a certain loss function. Specifically, $\boldsymbol{w}^*$ could be obtained by:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}} \boldsymbol{L}(\boldsymbol{w}) \tag{1}$$

In the analysis above, we consider the feature is represented in one-dimension. Now we extend the linear regression to multi-dimension situation when each label is dependant on d features. Consequently we have a feature set $\boldsymbol{A} \subseteq \mathbb{R}^{n \times d}$, and a label set $\boldsymbol{b} \in \mathbb{R}^n$. Here we propose two assumptions which stand immune through the lecture unless noted otherwise:

Assumption 1. For matrix $\boldsymbol{A} \subseteq \mathbb{R}^{n \times d}$, $n \gg d$ and $\boldsymbol{A}$ is full column rank.

Assumption 2. For each element in matrix $\boldsymbol{A}$ and $\boldsymbol{b}$, it takes $O(\log n)$ bits to store in the stream.

One could observe that the first assumption is actually a trivial one, considering n indicates the number of training samples, which is usually large. And there are no implications that the feature data samples follow a certain distribution or have explicit relationship between one another, hence each column should be linear independent to each other, it is therefore reasonable to regard $\boldsymbol{A}$ as full column rank. Now if we ignore the bias in linear equation for simplicity, then ideally, there exists $\boldsymbol{x} \subseteq \mathbb{R}^d$ *s.t.* $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$. Again we are trying to find an approximate $\tilde{\boldsymbol{x}}$. We adopt l_2*-norm loss function* as learning objective of this task, specifically as the following:

$$\begin{aligned} \boldsymbol{L}(\boldsymbol{x}) &= ||\boldsymbol{A} \cdot \boldsymbol{x} - \boldsymbol{b}||_2 \\ \boldsymbol{x}^* &= \underset{\boldsymbol{x}}{\operatorname{argmin}} \boldsymbol{L}(\boldsymbol{x}) \end{aligned} \tag{2}$$

Notice that equation 2 is a convex optimization problem, and there are several approaches to obtain $\boldsymbol{x}$. We provide the answer here as $\boldsymbol{x}^* = (A^T A)^{-1} A^T b$ and refer the readers to thinking of one possible proof using singular value decomposition.

The Model of Streaming l_2-Regression

In the streaming l_2-regression problem, on the contrary of observing and training all data samples as a family, we are receiving each data sample separately. Which is to say, we only have access to one row of matrix $\boldsymbol{A}, \boldsymbol{b}$ in each round. Can we design a streaming algorithm to find the exact $\boldsymbol{x}^*$ based on this?

The answer is no. One intuitive observation is that the overall space required for two matrices would be $O(n \cdot d \cdot \log n)$. If only a small portion of data samples are stored in the stream, then only the corresponding small portion in $\boldsymbol{x}^*$ could be learned. When the next round comes, the model would simply forget everything it has learned, and $\boldsymbol{x}^*$ goes back to blank. Notice that $\boldsymbol{x}^* \in \mathbb{R}^d$, so this only works when most elements in $\boldsymbol{x}^*$ are zero, or are dependant on very few features, which indicates most features do not have an impact on the label at all. So we claim that the exact $\boldsymbol{x}^*$ cannot be computed with $O(n)$ space, formally as claim 3. One could think of a rigorous proof by reduction from Index problem.

Claim 3. *The lower bound for exact l_2-regression problem in the stream is $\Omega(n)$.*

Now we consider solving it by approximation. Let's first define our problem:

Problem 1. Given each row of matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, denoted as $a_{i1}, a_{i2} \ldots a_{id}, (i \in [n])$, and corresponding row in matrix $\boldsymbol{b} \in \mathbb{R}^n$, denoted as b_i at each round, find $\tilde{\boldsymbol{x}}$ *s.t.* $\|\boldsymbol{A} \cdot \tilde{\boldsymbol{x}} - \boldsymbol{b}\|_2 \leqslant (1+\varepsilon) \cdot \|\boldsymbol{A} \cdot \boldsymbol{x}^* - \boldsymbol{b}\|_2$

We consider several natural streaming approaches and discuss how they may work as the following:

- **Uniform Sampling:** Sample a number of data points uniformly at random, compute $\boldsymbol{x}^*$ based on those samples and use $\boldsymbol{x}^*$ for $\tilde{\boldsymbol{x}}$. This is not promising to work for our case. Let's consider the situation illustrated in figure 1, most samples (blue) are in lower dimensions and a small number of samples (red) are in a distant, marginal dimension. If the uniform sampling happens to miss those red points, it is very likely that we would obtain l_1 instead of an approximation to l_2. An additional remark is that one could argue since the red points are outliers, it is possible that a robust regression model (l_1-regression for example) could work. The thought yields to further elaboration, with the focus on l_2-regression we should agree that the contribution of distant points can cause the approximation to go wild.
- **Leverage Score Sampling:** Define a score to indicate the importance of particular rows to the regression model, approximate the score based on some samplings. There may be ways to make this work, but we are not focusing on this approach in the lecture.
- **Linear Sketching:** The approach of linear sketching is usually applied as linear projection from high dimension to much lower dimension, but able to the properties of original vector with high probability. For our problem, we are using a matrix $S \in \mathbb{R}^{k \times n}$ to scale down matrix $A \in \mathbb{R}^{n \times d}$ to $\mathbb{R}^{k \times d}$. More specifically, we formulate linear sketching of l_2-regression problem as follows:

  Algorithm 1: Linear sketching for streaming l_2-regression

  1. Sample a matrix $S \in \mathbb{R}^{k \times n}$ from a certain distribution.
  2. For each set of data samples $(a_{i1}, a_{i2}, \ldots a_{id}, b_i)$, compute $S \cdot A^{(i)}$, $S \cdot b^{(i)}$ in the stream, where $A^{(i)}$ is padded with 0 to the shape of $n \times d$, and $b^{(i)}$ is padded with 0 to the shape of $n \times 1$.
  3. Solve $\tilde{x} = \underset{x \in \mathbb{R}^d}{\operatorname{argmin}} \|SAx - Sb\|_2$.

  Algorithm 1 is the backbone of the streaming algorithm of l_2-regression. The only task remained is how to sample a family of matrices $S \in \mathbb{R}^{k \times n}$ with 'small' k, we will return to that a bit later. Now let's analyze the space required in the linear sketching algorithm. We need to store matrix $S \in \mathbb{R}^{k \times n}$, $S \cdot A \in \mathbb{R}^{k \times d}$ and $S \cdot b \in \mathbb{R}^k$. It is worth of noticing that keeping one matrix of $S \cdot A$ will be enough to store

the all computation results, given the linearity that $S \cdot A = S \cdot (A^{(1)} + A^{(2)} + \ldots + A^{(n)}) = \sum_{i=1}^{n} S \cdot A^{(i)}$, the same works for $S \cdot b$. Therefore, it takes space of $O(k \cdot d \cdot \log n)$ to store $S \cdot A$ and $S \cdot b$. We are not covering how to store S in this lecture.

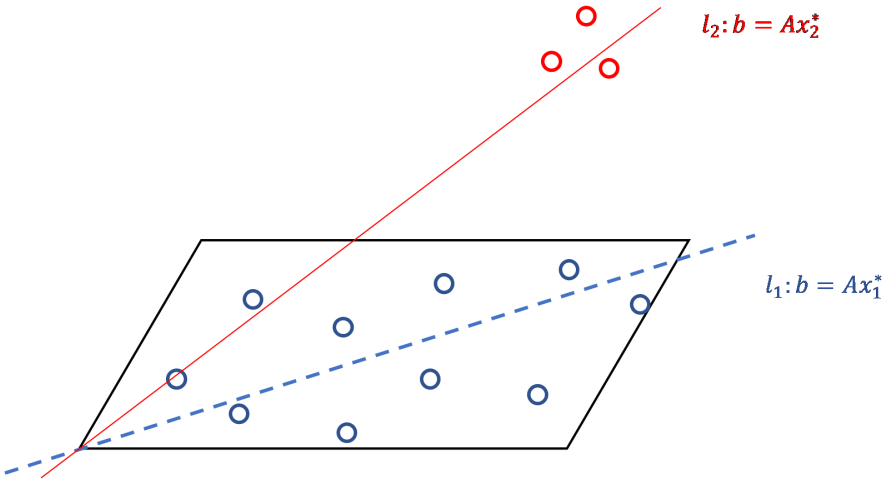

Figure 1: An example case that uniform sampling may not work

Let's now finalize the algorithm and prove the correctness. To provide a high-level idea, we design and prove the algorithm in three phases: (1) Dimensionality reduction from $n \times d$ to $k \times d$. (2) Subspace Embedding. (3) Regression. For the first step, any family of matrices having the property could be considered as candidate of S. For example, one may think of a matrix $S^{'} \in \mathbb{R}^{k \times n}$ with each element $S^{'}_{..} \in \{1, -1\}$ generated randomly. In this lecture we apply *Gaussian matrices* to solve this problem. Why could this help? Because in the i-th round of the algorithm, we only need the i-th row of A and i-th column of S, which can be generated from a Gaussian distribution with one step. Therefore, we only have to store k elements of each column in S, resulting in a space of $O(k \cdot \log n)$. If we can keep k small as $k \in poly(d)$, then the algorithm sounds good.

Before getting to the final version of the algorithm, we define Gaussian Matrix through a recap of Gaussian distribution.

Definition 4. A random variable X is said to have a Gaussian Distribution with mean $\mu \in R$ and variance $\sigma^2 \in R$, if its probability density function is given by:

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right) \tag{3}$$

Definition 5. A matrix $S \in \mathbb{R}^{m \times n}$ is a Gaussian matrix, if each element in S is sampled from a certain Gaussian distribution $N(\mu, \sigma^2)$, where μ is the mean and σ^2 is the variance.

Notice that Gaussian distribution has many favorable properties, and the correctness proof of our algorithm benefits a lot from these properties.

Finally, we propose the following algorithm:

Algorithm 2: An algorithm for streaming l_2-regression

1. Sample each element of S *i.i.d* $N(0, \frac{1}{k})$ for $k = O(\frac{d}{\varepsilon^2})$.
2. Run step 2 and 3 of Algorithm 1.

2 Proof of Correctness

We prove the correctness of this algorithm based on the following lemma.

Lemma 6. *For* $\tilde{x} \in \underset{x}{\operatorname{argmin}} \|SAx - Sb\|_2$, $\|A\tilde{x} - b\|_2 \leqslant (1 + 3\varepsilon) \cdot \|Ax^* - b\|_2$.

Notice that lemma 6 shows the output of Algorithm 2 satisfies the requirement of the original problem. For the rest of this lecture, we give the proof of lemma 6 in two steps:

1. If S is a subspace embedding, then the problem is solved. (informal version of claim 8)
2. Algorithm 2 will give us a subspace embedding S with $k = O(\frac{d}{\varepsilon^2})$ rows. (informal version of claim 9)

To lead out, we first introduce the notion of subspace embedding as following:

Definition 7. Given a matrix $M \in \mathbb{R}^{n \times d}$, a $(1 \pm \varepsilon)$-l_2 subspace embedding S of M is a matrix $S \in \mathbb{R}^{k \times n}$, such that $\forall y \in \mathbb{R}^d$, we have:

$$\|SMy\|_2 = (1 \pm \varepsilon)\|My\|_2$$

(Note: $a = (1 \pm \varepsilon)b$ means that $(1 - \varepsilon)b \leqslant a \leqslant (1 + \varepsilon)b$.)

One would observe that what subspace embedding does here is to preserve the norm of My within multiplicative approximation. From the perspective of measurement, the task is to find a linear transformation matrix S, such that the measurement of My stay in the bound of its multiplicative approximation. Therefore, it is a property of subspace $\mathbb{R}^k$ instead of $\mathbb{R}^n$, which means M could be a random matrix in its space. In consistence with the discussion of lemma 6, we have the following two claims to support our proof:

Claim 8 (Formalization of 1)**.** *Suppose* S *is a* $(1 \pm \varepsilon)$-l_2 *subspace embedding of* $\boldsymbol{M} \in \mathbb{R}^{n \times d}$*, then* $\|A\tilde{x} - b\|_2 \leqslant (1 + 3\varepsilon) \cdot \|Ax^* - b\|_2$

Claim 9 (Formalization of 2.)**.** *Subspace embedding problem can be solved for* $k = O(\frac{d}{\varepsilon^2})$

We present the proof of claim 9 in the next section and and the following goes goes the proof of claim 8:

Proof. Consider matrix $M = [A, b]$ and $y = [x, -1]^T$, where $A \in \mathbb{R}^{n \times d}, b \in \mathbb{R}^n$ and $x \in \mathbb{R}^d$, it is straightforward that $My = Ax - b$. Therefore, $\|SMy\|_2 = \|SAx - Sb\|_2$.

From the definition, we have $(1 - \varepsilon) \cdot \|Ax - b\|_2 \leqslant \|SAx - Sb\|_2 \leqslant (1 + \varepsilon) \cdot \|Ax - b\|_2$.

Recall that the output of algorithm 2 is $\tilde{x} \in \underset{x}{\operatorname{argmin}} \|SAx - Sb\|_2$, we have $\|A\tilde{x} - b\|_2 \leqslant (1+\varepsilon)\|SA\tilde{x} - Sb\|_2 \leqslant (1 + \varepsilon)\|SAx^* - Sb\|_2 \leqslant (1 + \varepsilon)^2\|Ax^* - b\|_2$.

Notice that ε is always expected to be small, let's say $\varepsilon \in [0, \frac{1}{2}]$, it is trivial to claim that $(1+\varepsilon)^2\|Ax^* - b\|_2 \leqslant (1 + 3\varepsilon)\|Ax^* - b\|_2$. ($(1 + \varepsilon)^2 = 1 + 2\varepsilon + \varepsilon^2 < 1 + 2\varepsilon + \varepsilon = 1 + 3\varepsilon$)

Concluding the proof. □

Subspace Embedding

For now, we are going to solve the subspace embedding problem (Claim 9). Just before that, we mention that we got an $O(k \cdot d \cdot \log n)$ space algorithm for regression, where $k = O(\frac{d}{\varepsilon^2})$. Instead of having a space of factor of n, it is now only $O(\frac{d}{\varepsilon^2})$, which is a much smaller space. First we will try to prove that $k = O(\frac{d}{\varepsilon^2})$. Then, for subspace embedding, we want to prove that for $\forall x \in \mathbb{R}^d$, $\|SAx\| = (1 \pm \varepsilon)\|Ax\|$. (For simplification, we use $\|a\|$ to indicate $\|a\|_2$ from now on.)

Simplifications

1. We can assume that columns of A are orthonormal:

Suppose $A = [A^1|A^2|\cdots|A^d]$, and $\|A^i\| = 1, < A^i, A^j >= 0$

The reason we can make such an assumption is as follows.

Claim 10. *Define $U \in \mathbb{R}^d$ s.t U has orthonormal columns, then we have two equivalent subspace $\{Uy|y \in \mathbb{R}^d\} = \{Ax|x \in \mathbb{R}^d\}$, for any $A \in \mathbb{R}^{n\times d}$ s.t A is full column rank.*

Because A is full column rank, so the columns of A is linearly independent, which means that A is a basis of $\mathbb{R}^d$ that spans the whole space. U has orthonormal columns, so its columns are also linearly independent, which means U is another basis of $\mathbb{R}^d$ that spans the whole space. Thus, we can prove that $\{Uy|y \in \mathbb{R}^d\}$ equals $\{Ax|x \in \mathbb{R}^d\}$. So we can claim that for each element in $\{Ax|x \in \mathbb{R}^d\}$, there exists an equivalent element in $\{Uy|y \in \mathbb{R}^d\}$.

Therefore we have:

$$\|SUy\| = (1 \pm \varepsilon)\|Uy\| \to \|SAx\| = (1 \pm \varepsilon)\|Ax\|,$$

which means that if we can prove the subspace embedding property for the left subspace, we can prove it for the right subspace as well. Therefore, we can just assume that the columns of our input A is orthonormal.

2. The other assumption is that we only need to focus on $\|x\| = 1$.

This is because for any other y, we can let $x = \frac{y}{\|y\|}$, then we have $\|x\| = 1$ and that $\|SAy\| = \|y\|\|SAx\|$.

At this point, A has orthonormal columns so we only care about $\|x\| = 1$.

Outline

With the simplifications mentioned above, we are going to prove claim 9 in two steps:

- Fix $x \in \mathbb{R}^d$ *s.t.* $\|x\| = 1$, and prove that $\|SAx\| = (1 \pm \varepsilon)\|Ax\|$, *w.p.* $1 - \frac{1}{2^{100d}}$
- Use Union bound to all $x \in \mathbb{R}^d$. (Note that this is not a trivial bound because $\mathbb{R}^d$ has infinite number of points, even if we constrain that $\|x\| = 1$.)

> **Remark.** How do we do union bound on infinite space S?
>
> The answer is $\gamma - net$ and we will explain this in details later.

Step 1

Given a Gaussian matrix $S \in \mathbb{R}^{k\times n}$ and an orthonormal matrix $A \in \mathbb{R}^{n\times d}$, we have $S \times A = G$, where $G \in \mathbb{R}^{k\times d}$. Recall that each element of S is sampled $i.i.d$ $N(0, \frac{1}{k})$ for $k = O(\frac{d}{\varepsilon^2})$. Then we have the following claim about G:

Claim 11. *G is a random Gaussian i.i.d[1] $N(0, \frac{1}{k})$ matrix.*

Proof. Recall that for Gaussian distribution, we have the following properties:

Property 1. Given $X \sim N(0, a^2)$ and $Y \sim N(0, b^2)$, we have $X + Y \sim N(a, a^2 + b^2)$

Property 2. If $g \sim N(0, \sigma^2)^{1\times n}$, $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$, and $\langle u, v\rangle = 0$, then $\langle g, u\rangle$ $\langle g, v\rangle$ are independent.

[1] independent and identically distributed

These properties can be used to prove claim 11. Recall that:

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} \begin{bmatrix} u_1 & u_2 & \cdots & u_3 \end{bmatrix} = \begin{bmatrix} \langle g_1, u_1 \rangle & \langle g_1, u_2 \rangle & \cdots \\ \langle g_2, u_1 \rangle & \cdots & \\ \vdots & & \end{bmatrix}$$

Where

(1)g_i is the rows of S, and $g_i \sim N(0, \frac{1}{k})^{1\times n}$ because S is a guassian matrix;

(2)u_i is the rows of A, and $u_i, u_j \in \mathbb{R}^n$ and $\langle u_i, u_j \rangle = 0$ for $i \neq j$ because A has orthonormal columns.

Next, we focus on the distribution of $\langle g_1, u_1 \rangle$. Note that $u_1 = (v_1, v_2, \cdots, v_n)$, $||u_1|| = 1$, then

$$\begin{aligned} \langle g_1, u_1 \rangle &= \textstyle\sum_{i=1}^n g_{1i} \cdot v_i \\ &= \textstyle\sum_{i=1}^n N(0, \frac{1}{k}) \cdot v_i \\ &= \textstyle\sum_{i=1}^n N(0, \frac{v_i^2}{k}) = N(0, \frac{\sum_{i=1}^n v_i^2}{k}) \\ &= N(0, \frac{1}{k}) \end{aligned}$$

Similarly, all other elements in G also satisfy the same distribution as $\langle g_1, u_1 \rangle$. Then we conclude the proof. □

The rest of Step 1 comes with the following claim:

Claim 12. *If x is a fixed vector such that $\|x\| = 1$, then $\|Gx\| = 1 \pm \varepsilon$.*

Proof. Let's focus on $\|Gx\|^2$ first, because if we can prove that $\|Gx\|^2$ is around $(1 \pm \varepsilon)$, then we can easily prove that $\|Gx\|$ is around $(1 \pm \varepsilon)$ as well. This is because $\varepsilon \in (0, \frac{1}{2}]$, thus $\sqrt{1+\varepsilon} < 1+\varepsilon$ and $\sqrt{1-\varepsilon} > 1-\varepsilon$. Note that $\|Gx\|^2 = \sum_{i=1}^k (\langle g_i, x \rangle)^2$, and that $\langle g_i, x \rangle \sim N(0, \frac{1}{k})$, then the question we care about becomes the following:

What is the value of $\sum_{i=1}^k [N(0, \frac{1}{k})]^2$?

Now we have k random variables and each of them is Gaussian $[N(0, \frac{1}{k})]^2$, and want to know what happens to their sum. To be more precise, define the following:

$$\begin{aligned} &Y_i \sim N(0, \tfrac{1}{k}), \\ &Z_i = Y_i^2, \\ &Z = \textstyle\sum_i Z_i, \end{aligned}$$

and we want to know $Pr(Z \in (1-\varepsilon, 1+\varepsilon))$. Note that:

$$E[Z] = \textstyle\sum_i Z_i = \sum_i [Y_i^2] = k \cdot \frac{1}{k} = 1.$$

($E[Y_i^2]$ just equals to the variance of Y_i because $Var[Y_i] = E[Y_i^2] - (E[Y_i]^2)$ and $E[Y_i] = 0$.)

So the problem has become a concentration question: Given that Z_i is independent, $Z_i \sim N(0, \frac{1}{k})^2$, $Z = \sum_i Z_i$, $E[Z] = 1$, what is $\Pr(|Z - E[Z]| > \varepsilon)$?

> **Remark.** Can we apply Chernoff bound here? Chernoff type bounds are false for $N(0, \frac{1}{k})^2$ so we need another type of bound here.

We introduce ***Bernstein inequality*** here:

Proposition 13. *Suppose* $W = \sum_{i=1}^{k} W_i$, $W_i \sim N(0,1)^2$, *then* $\forall t > 0$,

$$\Pr(|W - E[W]| > t + \sqrt{kt}) \leqslant e^{-\frac{t}{2}}.$$

Recall that we previously defined $Y_i \sim N(0, \frac{1}{k})$, $Z_i = Y_i^2$, $Z = \sum_i Z_i$.

Let's define $X_i = \sqrt{k} \cdot Y_i$, and $X_i \sim N(0,1)$. Define $X = \sum_{i=1}^{k} X_i^2$, and we can easily get $E[X] = k$ and $X = k \cdot Z$. Thus, $\Pr(|Z - E[Z]| > \varepsilon) = \Pr(|X - E[X]| > k\varepsilon)$. Apply Bernstein inequality to X and we have

$$\Pr(|X - k| > t + \sqrt{kt}) \leqslant e^{-\frac{t}{2}} < 2^{-\frac{t}{2}}$$

Let $t + \sqrt{kt} = k\varepsilon$ and $2^{-\frac{t}{2}} = \frac{1}{2^{100d}}$. Then $t = 200d$ and $200d + \sqrt{200d \cdot k} = k\varepsilon$. If we solve the equation, we have $k = O(\frac{d}{\varepsilon^2})$.

So far, we have proved that, for a single $x \in \mathbb{R}^d s.t. \|x\| = 1$, $\|SAx\| = (1 \pm \varepsilon)\|x\|$ w.p. $1 - 2^{-100d}$. □

Step 2: Union Bound

We cannot do a union bound over 2^{100d} vectors and get the type of bound we want. The problem is that we have infinite number of vectors. So it is not clear how to do union bound over infinite number of vectors. The fix for this issue is to use net arguments.

Let $B \subseteq \mathbb{R}^d$ be $B = \{x \mid \|x\| = 1\}$, for $0 < \gamma < 1$, a γ-*net* for B is

$$N \subseteq B,\ s.t\ \forall x \in B, \exists y \in N \cap ||x - y|| \leq \gamma.$$

What is a way of picking N?

Algorithm: Picking a γ-*net* N for B

1. Pick $y \in B$, add y to N, remove $\{x | x$ is in the γ ball of y$\}$ from B.
2. Recurse until B is empty.
3. Then return N is a y-*net*

Next, we have to get the size of N. Volume idea is a good way to bound the size of N. Now let's look at

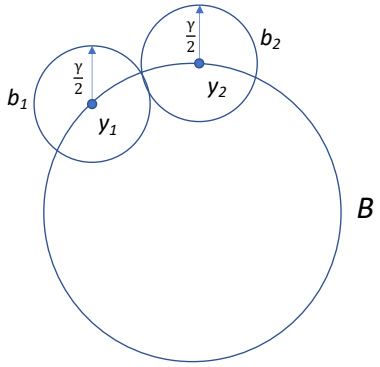

Figure 2: Example of using volume idea

the ball in figure 2. Suppose we have $y_1, y_2 \in N$, define b_i as the ball of radius $\frac{\gamma}{2}$ around y_i. Then we have the following properties:

- $b_i \cap b_j = \emptyset$ and $||y_i = y_j|| > \gamma$
- all b_i's are inside of a ball of radius $1 + \frac{\gamma}{2}$

The total volume of b_i's is less than the total volume of ball of radius $1 + \frac{\gamma}{2}$. Thus we have

$$||N|| \leq \frac{(1+\frac{\gamma}{2})^d}{(\frac{\gamma}{2})^d}$$

Later on the proof, we are going to pick γ as $\frac{1}{2}$, so for us $\gamma = \frac{1}{2} \rightarrow |N| \leq 5^d$.

Claim 14. *Fix a net N for B ($B = \{x|\|x\| = 1\}$), define $M = \{Ax|x \in N\}$, then $\forall x \in \mathbb{R}^d, \exists y \in M, s.t. \|Ax - y\| \leq \gamma$.*

Proof. To prove this, we need to use the fact that x is an orthonormal column.

$$\begin{aligned} x \in \mathbb{R}^d \rightarrow \exists z \in N \ s.t. \ \|Ax - Az\| \leq \gamma \\ \gamma \geq \|x - z\| = \|A(x - z)\| = \|Ax - Az\| \end{aligned} \tag{4}$$

We can find a $y \in M$ *s.t* $y = Az$, therefore we have $\gamma \geq \|Ax - y\|$. □

So far, we found this net inside the column space of A. Now we want to apply our dimension reduction lemma to all of $y \in M$ and all of their pariwise inner product. So we want to make the following claim.

Claim 15. *$\forall y_1, y_2 \in M$, the following must be true:*

1. $\|Sy_1\| = 1 \pm \varepsilon$
2. $\langle Sy_1, Sy_2 \rangle = \langle y_1, y_2 \rangle \pm O(\varepsilon)$, *w.p.* $1 - \frac{1}{2^d}$

Proof. Because $y_1 \in M$, $\exists x_1$ s.t. $y_1 = Ax_1$, we have

$$\|Sy_1\| = \|SAx_1\| = (1 \pm \varepsilon)\|Ax_1\|, \text{ w.p. } 1 - \frac{1}{2^{100d}}.$$

Since A is orthornormal and $\|x_1\| = 1$, $\|Ax_1\| = 1$, the first part is proven.

Now Let's prove the following proposition:

$$\|SA(x_1 - x_2)\|^2 = (1 \pm \varepsilon)\|A(x_1 - x_2)\|^2, \text{ w.p. } 1 - \frac{1}{2^{100d}}$$

We observe that:

$$\begin{aligned} \|SA(x_1 - x_2)\|^2 = \|SAx_1\|^2 + \|SAx_2\|^2 - 2\langle SAx_1, SAx_2 \rangle \\ \pm O(\varepsilon) \qquad \pm O(\varepsilon) \qquad \pm O(\varepsilon) \end{aligned}$$

Because we have $|M| = O(5^d)$, so we need a union bound over $5^{O(d)}$ pairs. We know this is true for y's $\in M$.

Look at y and we have

$$\exists y_1 \in M \text{ s.t } \|y - y_1\| \leq \gamma.$$

Suppose $\|y - y_1\| = \alpha$, and define $y' = \frac{y - y'}{\alpha}$, then $\|y_1'\| = 1$, which means that

$$\exists y_2' \text{ s.t } \|y_1' - y_2'\| \leq \gamma.$$

Now let's define $y_2 = \alpha y_2'$, then

$$\|y_1' - y_2'\| = \|\tfrac{y-y_1}{\alpha} - \tfrac{y_2}{\alpha}\| \leq \alpha \longrightarrow \|y - y_1 - y_2\| \leq \alpha \cdot \gamma \leq \gamma^2$$

(Note that $\alpha \leq \gamma$.)

Continue for i steps, we have:

$$\|y - (y_1 + y_2 + \cdots + y_i)\| \leq \gamma^i$$

Therefore, we show that $\|y_i\| \leq r^{i-1}$.

Finally, let's extend i to infinity. For $i = \infty$, define $y = \sum_i y_i$, then

$$\begin{aligned}
\|Sy\|^2 &= \|S(\sum_i y_i)^2\| \\
&= \sum_i \|Sy_i\|^2 + \sum_{i \neq j} 2\langle Sy_i, Sy_j \rangle \\
&= \sum_i (1 \pm \varepsilon)\|y_i\|^2 + \sum_{i \neq j} (2\langle y_i, y_j \rangle \pm O(\varepsilon) \cdot \|y_i\|\|y_j\|) \\
&= \|\sum y_i\|^2 \pm \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} O(\varepsilon) \cdot \|y_i\| \cdot \|y_j\| \\
&= 1 \pm O(\varepsilon) \cdot \sum_{i=1}^{\infty} \|y_i\| \cdot \big(\sum_{j=1}^{\infty} \|y_j\|\big) \\
&= 1 \pm O(\varepsilon)
\end{aligned} \tag{5}$$

□

To this end, we have proved the correctness of Algorithm for Streaming Linear Regression. We hope figure 3 could help understand the structure of the whole proof.

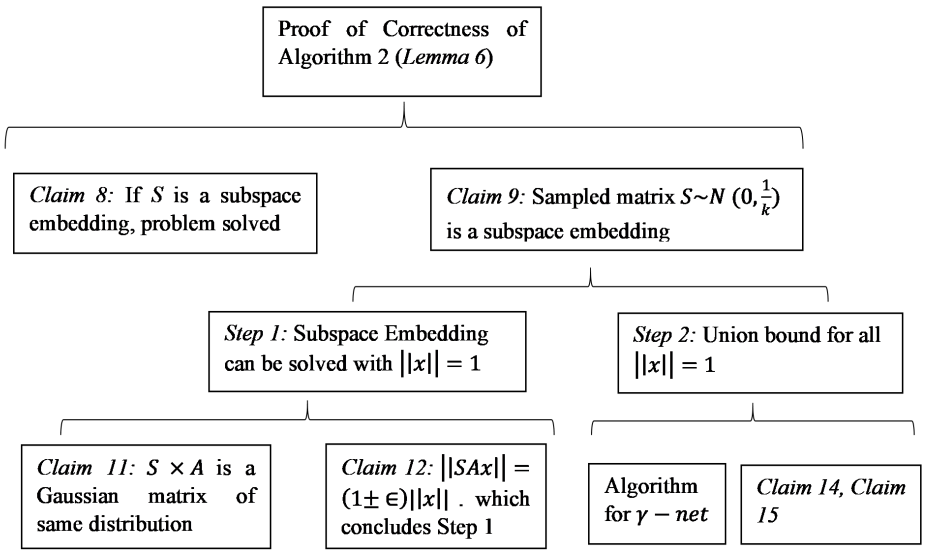

Figure 3: Structure of Correctness Proof of Algorithm 2