COMS E6998-9: Algorithmic Techniques for Massive Data — Oct 6, 2015

# Lecture 9 – Fast Dimension Reduction

Instructor: *Alex Andoni* — Scribe: *Negev Shekel Nosatzki*

## 1 Johnson-Lindenstrauss Summary

- $F(x) = \frac{1}{\sqrt{k}} G_{k*d} x$
- $\|F(x)\| = (1 \pm \epsilon)\|x\|$ with probability $\geq 1 - \delta$
- $k = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$
- Takes time $O(k \cdot d)$ as we need to calculate k*d dense matrix

## 2 Fast Johnson-Lindenstrauss Transformation Idea and Issues

### 2.1 Running Time Goal

- $O(d + k)$ is optimal goal
- We'll show $O(d \log d + k^3)$

### 2.2 Sampling

To improve the algorithm speed, we can sample s entries from each row. We can define:

- $h : [d] \to \{0, 1\}$
- $\Pr[h(i) = 1] = \frac{s}{d}$

And compute:

- $z = \sqrt{\frac{d}{s}} \sum_{i=1}^{d} h(i) \cdot g_i x_i$
- $\mathbb{E}[\|z\|^2] = \frac{d}{s} \mathbb{E}[\sum_{i=1}^{k} h(i) \cdot g_i^2 x_i^2] = \|x\|^2$

While this tactic works when x is dense, x can be sparse which can create large variance.

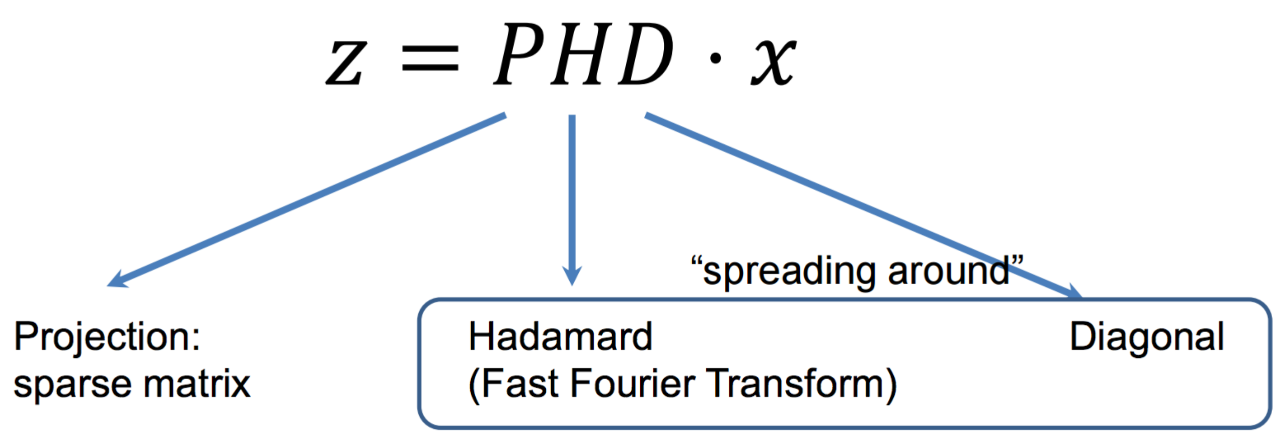
### 2.3 Example of sparse x

Consider the case where $x = e_1 - e_2 \implies$ even choosing relatively large sample size $s = \frac{d}{k}$ has high chance to fail since $\Pr[h(1) = 1 \wedge h(2) = 1] = (\frac{s}{d})^2 = \frac{1}{k^2}$.
And since we have k rows the overall chance is $\frac{1}{k}$ which is too high.

### 2.4 Spreading x

To solve the above issue we will "spread-around" x and use sparse G.

## 3 FJLT construction




### 3.1 Spreading x into y - Overview

The idea is to spread x into y, by defining $y = HDx$. y is in dimension d (like x) and $\|y\| = \|x\|$. However, unlike x, we will be able to provide certain guarantees as to the maximum coordinate values, and therefore we can project y into lower-dimensional z using a sparse matrix P with high probability.

### 3.2 Definitions

- D = diagonal matrix with random $\pm 1$ on diagonal
- H = Hadamard Matrix = Fourier Transform
- P = Projection Matrix - similar to previous G but sparse and dimension $k' * d$, with $k' \approx k^2$

### 3.3 Why Fourier Transform?

Fourier Transform is non-trivial rotation. A trivial rotation (i.e. random) takes $O(d^2)$ to compute, while FT takes $O(d \log d)$.

$$H_1 = 1$$

$$H_{2^l} = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{2^{l-1}} & H_{2^{l-1}} \\ H_{2^{l-1}} & -H_{2^{l-1}} \end{pmatrix}$$

$$H_{d*d} = \begin{pmatrix} H_1 \\ H_2 \\ ... \\ H_i \\ ... \\ H_d \end{pmatrix}$$

Where $H_{ij} = \pm \frac{1}{\sqrt{d}}$.
Therefore, $y_i = H_i D x = rx$, where $rx$ is a random vector of $\pm \frac{1}{\sqrt{d}}$

**Lemma 1.** *$r \cdot x$ behaves like $g \cdot x$*

This needs to be proved (wasn't proved in class). Also, we need to bound $y_i$.

**Lemma 2.** $\Pr[y_i^2 \leq \frac{1}{d} \cdot O(\log \frac{1}{\delta})] \geq 1 - \delta$

*Proof.* We will approximate $y_i \approx g \cdot x \sim l$ where l is Gaussian $\implies \frac{1}{\sqrt{2\pi}} \cdot e^{\frac{-l^2}{2}} < \delta$ when $l \approx \sqrt{\log \frac{1}{\delta}}$ □

### 3.4 Why do we need D?

If x is sparse, then $Hx$ is dense. However $\exists$ dense $x$ s.t. $Hx$ is sparse. $D$ fixes it by randomizing $H$ ($HD$ is randomization of $H$) and since there are very few such dense $x$, randomization fixes that issue.

### 3.5 $y_i$ Dependence - issue?

Clearly, $y_i$ are not independent:

- $y_1 = H_1 D x$
- $y_2 = H_2 D x$
- and so on.

However, since we are only rotating, the norm doesn't change: $\|y\| = \|x\|$!

# 4 P Projection

## 4.1 Density of y

As we saw: $y_i^2 \leq \frac{1}{d} \cdot O(\log \frac{1}{\delta})$ with prob. $1 - \delta$; and since y has d coordinates, we get:

$$m = \max y_i^2 \leq \frac{1}{d} \cdot O(\log \frac{1}{\delta}) \text{ with prob. } 1 - d\delta \Longrightarrow \tag{1}$$

$$m \leq \frac{1}{d} \cdot O(\log \frac{d}{\delta}) \text{ with prob. } 1 - \delta \tag{2}$$

## 4.2 Projecting to z

Define:

- $j \in [k']$
- $z_j = y_i$ for random $i \in [d] \rightarrow \forall i, j; \Pr[z_j = y_i] = \frac{1}{d}$
- Assume w.l.o.g $\|x\| = 1$

**Claim 3.** $\|z\|^2 = (1 \pm \epsilon)\|x\|^2$ *with prob.* $1 - 2\delta$

We want to show $\sum_j z_j^2$ concentrates.

Define:

- $t_j = \frac{z_j^2}{m} \in [0, 1]$
- $\mu = \mathbb{E}[\sum_{j=1}^{k'} t_j]$

*Proof.*

$$\mu = \mathbb{E}[\sum_j \frac{z_j^2}{m}] = \frac{1}{m} \sum_j [\frac{1}{d} y_1^2 + \frac{1}{d} y_1^2 + ...] = \frac{1}{md} \sum_j \|y\| = \frac{k'}{md} \Longrightarrow \tag{3}$$

$$\text{Chernoff:} \Pr[\sum_j t_j \notin (1 \pm \epsilon)\mu] \leq 2e^{\frac{-\epsilon^2 \mu}{3}} = 2e^{\frac{\epsilon^2 k'}{3md}} < \delta \Longrightarrow \tag{4}$$

$$k' = m \cdot d \cdot \frac{3}{\epsilon^2} \cdot \ln \frac{2}{\delta} = O(\log \frac{d}{\delta} \cdot \frac{1}{e^2} \cdot \log \frac{1}{\delta}) \tag{5}$$

Since each of Chernoff and m can deviate from bound with prob. $\delta$, the overall success rate is $1 - 2\delta$. ☐

# 5 Time analysis and further reduction

So far we reduced dimension d to k' with time $O(d \log d + k')$:

- $d \log d \to HDx$ multiplication
- $k' \to$ Projection

To further reduce dimension from $k'$ to $k = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$, we can apply regular (dense) JL on z:

- $Gz$ projection takes $k' \cdot k$ time.
- Final time for $d \to k$ dimension reduction: $O(d \log d + k \cdot k') = O(d \log d + k^3)$

## 5.1 Example

Assume:

- $d = \log^3 n$
- $\delta = \frac{1}{n^2}$

We get:

$$k = O(\frac{1}{\epsilon^2} \log n) \tag{6}$$

$$k' = O(\frac{1}{\epsilon^2} \log^2 n) \tag{7}$$

$$\text{FJL Time} : O(\log^3 n \log \log n + \frac{1}{\epsilon^4} \log^3 n) \tag{8}$$

$$\text{JL Time} : O(dk) = O(\frac{1}{\epsilon^2} \log^4 n) \tag{9}$$

Since we assume $\epsilon$ is constant $\Rightarrow$ FJL Time $\ll$ JL Time.

## 5.2 Optimal time

What can we hope for?

- $O(d + k)$ or $O(d \log d + k)$
- Assume $d = \log n$
- JL Time: $O(dk) \approx \log^2 n$
- Optimal Time: $O(d + k) \approx \log n$