Lecture 15: Dimensionality Reduction

John Sylvester   Nicolás Rivera   Luca Zanetti   Thomas Sauerwald

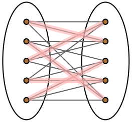
Lent 2019

Outline

Mathematical Tools

- **Matrices and Geometry**
  - Data points (predictions, observations, classifications) encoded in matrices/vectors
  - This allows geometric representation that is the basis of many network analysis methods (e.g., clustering)
  - Networks and graphs ⇔ adjacency matrices

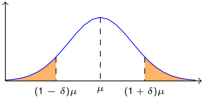
  Inner product, Hyperplanes, Eigenvectors



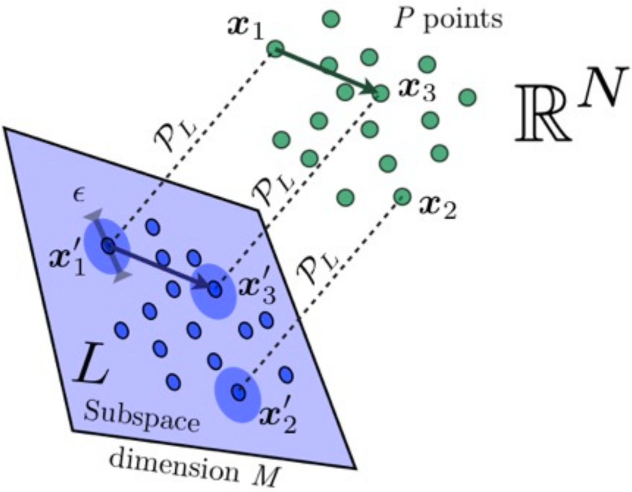
$$A = \begin{pmatrix} 0&0&0&0&0&1&1&0&0&1\\ 0&0&0&0&0&0&1&1&0&1\\ 0&0&0&0&0&1&1&0&1&0\\ 0&0&0&0&0&1&0&1&1&0\\ 0&0&0&0&0&0&0&1&1&1\\ 1&0&1&1&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&0&0&0\\ 0&1&0&1&1&0&0&0&0&0\\ 0&0&1&1&1&0&0&0&0&0\\ 1&1&0&0&1&0&0&0&0&0 \end{pmatrix}$$

- **Probability Theory**
  - Randomisation guards against worst-case inputs
  - Sampling allows approximate answers/estimates without looking at entire input
  - Random Projection is a powerful preprocessing tool to compress data using redundancy
  - Randomised Algorithms often exploit concentration

  Random Variables, Chernoff Bounds, hashing



Motivation: Dimensionality Reduction



Source: Laurent Jacques

Motivation

Random Projection:

- Powerful technique for performing dimensionality reduction
- Theoretical guarantee given by Johnson-Lindenstrauss Lemma
- Key Idea: Compress data set (set of vectors) through multiplication with a random matrix

We will **first** look at a simpler algorithm that instead involves multiplying matrices by a random vector!

Outline

Matrix Multiplication

Remember: If $A = (a_{ij})$ and $B = (b_{ij})$ are square $n \times n$ real-valued matrices, then the matrix product $C = A \cdot B$ is defined by

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj} \qquad \forall i, j = 1, 2, \ldots, n.$$

- Naive Algorithm: $O(n^3)$ time
- Strassen's Algorithm (1969): $O(n^{2.81})$ time
- State-of-the-art: Williams, Virginia Vassilevska (2013): $O(n^{2.3729})$ time

Remarkable: It is possible to verify matrix multiplication in $O(n^2)$!

R. M. Freivalds (1942-2016), Latvian computer scientists and mathematician

Source: Wikipedia

Freivalds' Algorithm

Freivalds' Algorithm

Input: Three $n \times n$ matrices A, B and C

1. Sample a random $\{0, 1\}$-vector $r = (r_1, r_2, \ldots, r_n)$
2. Compute a new vector $p = (AB - C)r = A(Br) - Cr$
3. If $p \neq \vec{0}$, then REJECT.
4. Otherwise, ACCEPT.

Important to keep running time within $O(n^2)$!

Example of a one-sided error!

Correctness Analysis

- If $AB = C$, then Freivalds' Algorithm returns ACCEPT.
- If $AB \neq C$, then Freivalds' Algorithm returns REJECT w.p. $1/2$.

Running the algorithm k times reduces the error probability to $(1/2)^k$!

Proof of Correctness

- Consider the (only non-trivial) case when $AB \neq C$. We need to prove that:

$$\mathbf{P}\left[\, p = \vec{0} \,\right] \leq 1/2.$$

- Define a new matrix $D = A \cdot B - C$
- At least one element D is nonzero; call this d_{ik}
- Then, element p_i is obtained by

$$p_i = \sum_{j=1}^{n} d_{ij} r_j = d_{ik} r_k + \sum_{j \neq k} d_{ij} r_j$$

- Let $\sum_{j \neq i} d_{ij} r_j =: \alpha$ for some random variable $\alpha \in \mathbb{R}$.
- Using Bayes' rule gives:

$$\begin{aligned}
\mathbf{P}[\, p_i = 0 \,] &= \mathbf{P}[\, p_i = 0 \mid \alpha = 0 \,] \cdot \mathbf{P}[\, \alpha = 0 \,] + \mathbf{P}[\, p_i = 0 \mid \alpha \neq 0 \,] \cdot \mathbf{P}[\, \alpha \neq 0 \,] \\
&\leq \mathbf{P}[\, r_k = 0 \mid \alpha = 0 \,] \cdot \mathbf{P}[\, \alpha = 0 \,] + \mathbf{P}[\, r_k = 1 \mid \alpha \neq 0 \,] \cdot \mathbf{P}[\, \alpha \neq 0 \,] \\
&\leq \frac{1}{2} \cdot \mathbf{P}[\, \alpha = 0 \,] + \frac{1}{2} \cdot \mathbf{P}[\, \alpha \neq 0 \,] = \frac{1}{2}. \qquad \square
\end{aligned}$$

Comments on Freivalds' Algorithm

- Why do we choose each entry of r from $\{0,1\}$ uniformly at random?
  - This allows algorithm to work in $\mathbb{F}_2$ (the "smallest" field)
  - Choosing each entry from $\{0,1,\ldots,x-1\}$ increases probability for REJECT if $AB \neq C$ is increased to $1-1/x$.

- How can we reduce the probability of error?
  - Run Freivalds' k times and REJECT if at least one of run returns REJECT

  $\Rightarrow$ The probability for REJECT if $AB \neq C$ is increased to $1-(1/2)^k$.

- Can we find an efficient deterministic algorithm to verify Matrix Multiplication?
  - This is a fundamental open problem. (Even if it was possible, it is likely that the algorithm would be much more complicated!)
  - Note: For any deterministic vector r, it is easy to find matrices A, B and C so that $(AB-C)\cdot r = 0$ but $AB \neq C$!

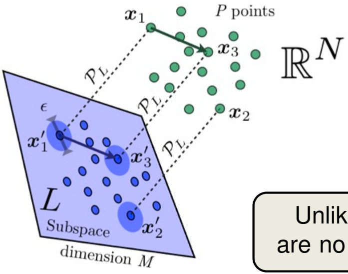
    Proof: Let $D = AB - C$.
    If $r = \vec{0}$, then we can choose D differently from the all zero-matrix.
    Otherwise, let $r_k \neq 0$, and then for any $1 \leq i \leq n$:

$$\sum_{j=1}^{n} d_{ij} r_j = 0 \quad \Leftrightarrow \quad d_{ik} r_k = \sum_{j \neq k} d_{ij} r_j \quad \Leftrightarrow \quad d_{ik} = \frac{\sum_{j \neq k} d_{ij} r_j}{r_k}.$$

    Now choose all $d_{ij} \neq 0, j \neq k$ arbitrarily and then pick d_{ik} to solve above equation.

Outline

Dimensionality Reduction: Basic Setup



Unlike other methods like PCA, there are no assumptions on the original data.

- Given P points $x_1, x_2, \ldots, x_P \in \mathbb{R}^N$
- Want to find P points $x'_1, x'_2, \ldots, x'_P \in \mathbb{R}^M$, $M \ll N$

Goal: Distances are approximately preserved, i.e.,

$$(1-\epsilon) \cdot \|x_i - x_j\| \leq \|x'_i - x'_j\| \leq (1+\epsilon) \cdot \|x_i - x_j\| \qquad \text{for all } i, j$$

Johnson-Lindenstrauss-Lemma

Note: M does not depend on N!

Theorem

Let $x_1, x_2, \ldots, x_p \in \mathbb{R}^N$ be arbitrary. Pick any $\epsilon = (0,1)$. Then for some $M = O(\log(P)/\epsilon^2)$, there is a polynomial-time algorithm that, with probability at least $1 - \frac{2}{P}$, computes $x_1', x_2', \ldots, x_P' \in \mathbb{R}^M$ such that

$$(1-\epsilon)\cdot\|x_i - x_j\| \leq \|x_i' - x_j'\| \leq (1+\epsilon)\cdot\|x_i - x_j\| \qquad \text{for all } i,j$$
$$(1-\epsilon)\cdot\|x_i\| \leq \|x_i'\| \leq (1+\epsilon)\cdot\|x_i\| \qquad \text{for all } i.$$

How to construct $x_1', x_2', \ldots, x_P'$?

Key Tool: Random Projection Method

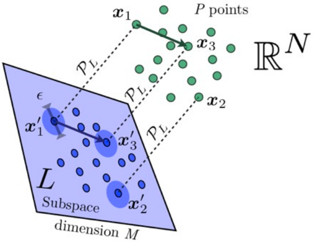
Definition of $f : \mathbb{R}^N \to \mathbb{R}^M \qquad (M \ll N)$

$$f\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} \cdots\cdots r_1^T \cdots\cdots \\ \cdots\cdots r_2^T \cdots\cdots \\ \vdots \\ \cdots\cdots r_M^T \cdots\cdots \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} r_1^T w \\ r_2^T w \\ \vdots \\ r_M^T w \end{pmatrix},$$

where the r_i's are random

Each entry of r_i is independently drawn from $\mathcal{N}(0,1)$

r_i's are chosen independently



Johnson-Lindenstrauss Lemma

Let $w \in \mathbb{R}^N$ with $\|w\| = 1$. Then for some $M = O(\log(P)/\epsilon^2)$, we have

$$\mathbf{P}\left[1 - \epsilon \leq \frac{\|f(w)\|}{\sqrt{M}} \leq 1 + \epsilon\right] \geq 1 - \frac{2}{P^3}.$$

Proof of Theorem (using JL-Lemma)

Johnson-Lindenstrauss Lemma

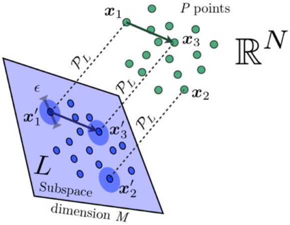
Let $w \in \mathbb{R}^N$ with $\|w\| = 1$. Then for some $M = O(\log(P)/\epsilon^2)$, we have

$$\mathbf{P}\left[1 - \epsilon \leq \frac{\|f(w)\|}{\sqrt{M}} \leq 1 + \epsilon\right] \geq 1 - \frac{2}{P^3}.$$

- Define $L(v) := \frac{f(v)}{\sqrt{M}}$
- JL-Lemma with $w = \frac{v}{\|v\|} \Rightarrow \frac{\|f(w)\|}{\sqrt{M}} = \frac{\|L(v/\|v\|)\sqrt{M}\|}{\sqrt{M}}$

$$\mathbf{P}[\,(1-\epsilon)\cdot\|v\| \leq \|L(v)\| \leq (1+\epsilon)\cdot\|v\|\,] \geq 1 - \frac{2}{P^3}.$$

- Apply to $v = x_j$ and $v = x_i - x_j, j \neq i$ and the Union bound ($\mathbf{P}[\,A \cup B\,] \leq \mathbf{P}[\,A\,] + \mathbf{P}[\,B\,]$): W.p. $1 - \frac{2}{P}$,



$L(x_i - x_j) = L(x_i) - L(x_j)$

$$(1-\epsilon)\cdot\|x_i - x_j\| \leq \|L(x_i - x_j)\| \leq (1+\epsilon)\cdot\|x_i - x_j\| \qquad \text{for all } i, j$$

$$(1-\epsilon)\cdot\|x_i\| \leq \|L(x_i)\| \leq (1+\epsilon)\cdot\|x_i\| \qquad \text{for all } i.$$

□

Example: Target Dimension M of Dimensionality Reduction

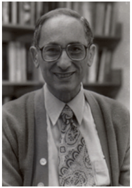
Recall: $M \leq \frac{6 \ln P}{\epsilon^2}$

ϵ	Number of Points P	Target Dimension M
1/2	1,000	166
1/2	10,000	221
1/2	100,000	276
1/2	1,000,000	331
1/2	10,000,000	387
1/10	1,000	4145
1/10	10,000	5526
1/10	100,000	6907
1/10	1,000,000	8298
1/10	10,000,000	9670

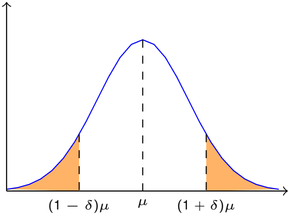
Outline

Reminder: Chernoff Bounds

- Chernoffs bounds are “strong” bounds on the tail probabilities of sums of independent random variables (random variables can be discrete or continuous)
- usually these bounds decrease exponentially as opposed to a polynomial decrease in Markov’s or Chebysheff’s inequality (see example later)
- have found various applications in:
  - Random Projections
  - Approximation and Sampling Algorithms
  - Learning Theory (e.g., PAC-learning)
  - Statistics
  - ⋮



Hermann Chernoff (1923-)



Recipe for Deriving Chernoff Bounds

Recipe

The three main steps in deriving Chernoff bounds for sums of independent random variables $X = X_1 + \cdots + X_n$ are:

1. Instead of working with X, we switch to $e^{\lambda X}$, $\lambda > 0$ and apply Markov's inequality $\rightsquigarrow \mathbf{E}\left[e^{\lambda X} \right]$
2. Compute an upper bound for $\mathbf{E}\left[e^{\lambda X} \right]$ (using independence of $X_1, \ldots, X_n$)
3. Optimise value of λ to obtain best tail bound

Outline

Proof of JL-Lemma (1/4)

Johnson-Lindenstrauss Lemma

Let $w \in \mathbb{R}^N$ with $\|w\| = 1$. Then for some $M = O(\log(P)/\epsilon^2)$, we have

$$\mathbf{P}\left[1 - \epsilon \leq \frac{\|f(w)\|}{\sqrt{M}} \leq 1 + \epsilon\right] \geq 1 - \frac{2}{P^3}.$$

Proof (of the upper bound):

- Squaring yields $\mathbf{P}\left[\ \|f(w)\|^2 > (1+\epsilon)^2 \cdot M\ \right]$.
- Recall that the i-th coordinate of $f(w)$ is $r_i^T \cdot w$. The distribution is

$$\mathcal{N}(0, \sum_{j=1}^{N} w_j^2) = \mathcal{N}(0, 1).$$

If $X_1, \ldots, X_N$ are independent random variables with distribution $\mathcal{N}(0,1)$ each, then $\sum_{j=1}^N w_j X_j$ has distribution $\mathcal{N}(0, \sum_{j=1}^N w_j^2)$

- Hence

$$\|f(w)\|^2 = \sum_{i=1}^{M} X_i^2,$$

where the X_i's are independent $\mathcal{N}(0,1)$ random variables.

Proof of JL-Lemma (2/4)

- Taking expectations:

$$\mathbf{E}\left[\|f(w)\|^2\right] = \mathbf{E}\left[\sum_{i=1}^{M} X_i^2\right]$$
$$= \sum_{i=1}^{M} \mathbf{E}\left[X_i^2\right] = M$$

- We will now derive a Chernoff bound for $X := \sum_{i=1}^{M} X_i^2$. Let $\lambda \in (0, 1/2)$,

$$\mathbf{P}[X > \alpha] = \mathbf{P}\left[e^{\lambda Y} > e^{\lambda\alpha}\right] \leq e^{-\lambda\alpha} \cdot \mathbf{E}\left[e^{\lambda X}\right].$$

- Since $X_1^2, \ldots, X_M^2$ are independent,

$$\mathbf{E}\left[e^{\lambda X}\right] = \mathbf{E}\left[e^{\lambda \sum_{i=1}^{M} X_i^2}\right] = \mathbf{E}\left[\prod_{i=1}^{M} e^{\lambda X_i^2}\right] \stackrel{!}{=} \prod_{i=1}^{M} \mathbf{E}\left[e^{(\lambda X_i^2)}\right]$$

Proof of JL-Lemma (3/4)

- We need to analyse $\mathbf{E}\left[\, e^{\lambda X_i^2} \,\right]$:

$$\mathbf{E}\left[\, e^{\lambda X_i^2} \,\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(\lambda y^2) \exp(-y^2/2) dy$$
$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(-y^2(1-2\lambda)/2\right) dy$$

- Now substitute $z = y \cdot \sqrt{1-2\lambda}$ to obtain

$$= \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{1-2\lambda}} \cdot \int_{-\infty}^{\infty} e^{-z^2/2} dz$$
$$= \frac{1}{\sqrt{1-2\lambda}}$$

$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-z^2/2} dz$ is the CDF of $\mathcal{N}(0,1)$

Proof of JL-Lemma (4/4)

- Hence with $\alpha = (1+\epsilon)^2 M$,

$$\mathbf{P}\left[X > (1+\epsilon)^2 M\right] \leq e^{-\lambda(1+\epsilon)^2 M} \cdot \left(\frac{1}{1-2\lambda}\right)^{M/2}$$

- We choose $\lambda = (1 - 1/(1+\epsilon)^2)/2$, giving

$$\mathbf{P}\left[X > (1+\epsilon)^2 M\right] \leq e^{(M - M(1+\epsilon)^2)/2} \cdot (1+\epsilon)^{-M}$$

- The last term can be rewritten as

$$\exp\left(\frac{M}{2}\left(1-(1+\epsilon)^2\right) - \frac{M}{2}\ln\left(\frac{1}{(1+\epsilon)^2}\right)\right)$$
$$= \exp\left(-M\left(\epsilon + \epsilon^2/2 - \ln(1+\epsilon)\right)\right)$$

- Using $\ln(1+x) \leq x$ for $x \geq 0$, implies

$$\mathbf{P}\left[X > (1+\epsilon)^2 M\right] \leq \exp\left(-M\left(\epsilon + \epsilon^2/2 - \epsilon\right)\right)$$
$$\leq \exp\left(-M\epsilon^2/2\right).$$

- With $M = 6 \ln P/\epsilon^2$, the last term becomes $\frac{2}{P^3}$.
- Lower bound is derived similarly $\Rightarrow$ proof complete $\square$

Outline

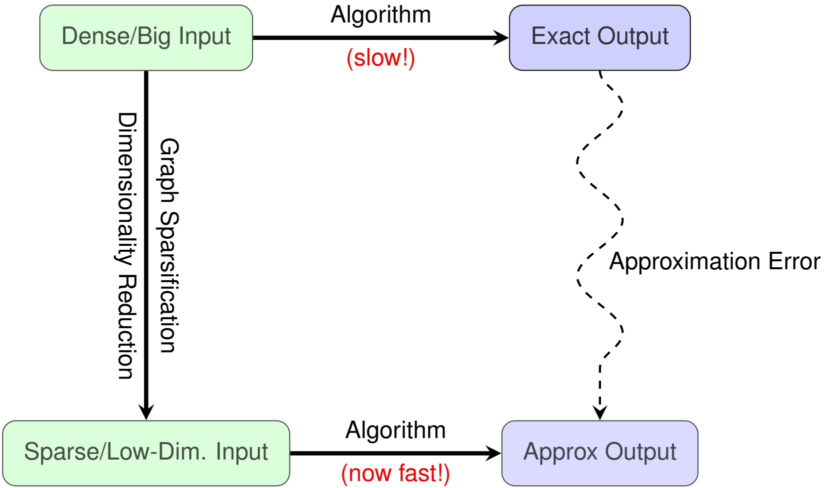
General Comments on the JL-Lemma

- Use Random Projection to a Subspace
  - similar to projection on the bottom k eigenvectors, but with different aim here
  - exploits redundancy in “Wide-Data” (high-dimensional data)
  - also powerful method in approximation algorithms (recall SPD algorithm for MAX-CUT!)

- Why do we use a Random Projection?
  - If projection f is chosen deterministically, easy to find vectors u, v with $\|u - v\|$ large but $f(u) = f(v)$.

  $\Rightarrow$ Randomisation prevents the input to foil a specific deterministic algorithm

Generic Application of Preprocessing



Further Reading (1/2)

Is the dependence on the dimension optimal?

- This had been an open problem for many years
- Theoretical results eventually established that the dependence is basically optimal (see research articles for more details)

"Database-Friendly" Version of JL

- Random Matrix contains only three values: $\{-1, 0, +1\}$

Applications of JL in Streaming

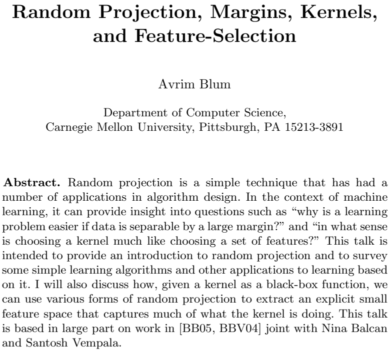
- many streaming algorithms based on JL
- one basic example is to estimate the frequencies
- often use projections based on sparse matrices which have a succinct representation

Further Reading (2/2)

Applications of JL in Machine Learning

- Streaming Algorithms
- Preprocessing of many Machine Learning Methods like Clustering

. . .



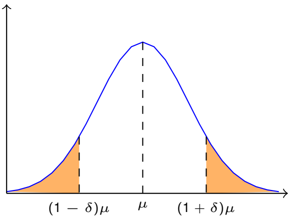
Random Projection, Margins, Kernels, and Feature-Selection

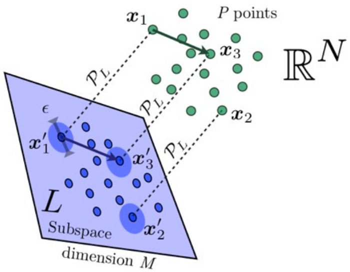
Avrim Blum

Department of Computer Science,
Carnegie Mellon University, Pittsburgh, PA 15213-3891

Abstract. Random projection is a simple technique that has had a number of applications in algorithm design. In the context of machine learning, it can provide insight into questions such as "why is a learning problem easier if data is separable by a large margin?" and "in what sense is choosing a kernel much like choosing a set of features?" This talk is intended to provide an introduction to random projection and to survey some simple learning algorithms and other applications to learning based on it. I will also discuss how, given a kernel as a black-box function, we can use various forms of random projection to extract an explicit small feature space that captures much of what the kernel is doing. This talk is based in large part on work in [BB05, BBV04] joint with Nina Balcan and Santosh Vempala.

Summary: Using Chernoff Bounds for Dimensionality Reduction





- sums of independent random variables
- Chernoff Bounds: concrete tail inequalities that are exponential in the deviation
- Proof Method: Moment Generating Function & Markov's Inequality

- Random Projection Method
  - multiply by a random matrix
  - preserves distances up to $1 \pm \epsilon$
  - new dimension $\mathcal{O}(\log P/\epsilon^2)$