Linear Algebra Recap

Anton Tsitsulin, Davide Mottin

Hasso Plattner Institute

Graph Mining course Winter Semester 2017

Vectors and norms

- Vector is a 1D array with n numbers
- Main questions:
  - **How similar** two vectors are?
  - **How good** our approximation of the vector is?

- **Norm** of the vector $||x||$ is a qualitative measure of *size* such that:
  - $||\alpha x|| = |\alpha| \cdot ||x||$
  - $||x + y|| \leq ||x|| + ||y||$ triangle inequality
  - If $||x|| = 0$ then $x = 0$
- The distance between two vectors x and y is defined as:
  - $d(x, y) = ||x - y||$

Standard norms

- The most well-known is **Euclidean norm:**

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$$

Quiz: Why is the modulo sign here?

A: x may be complex

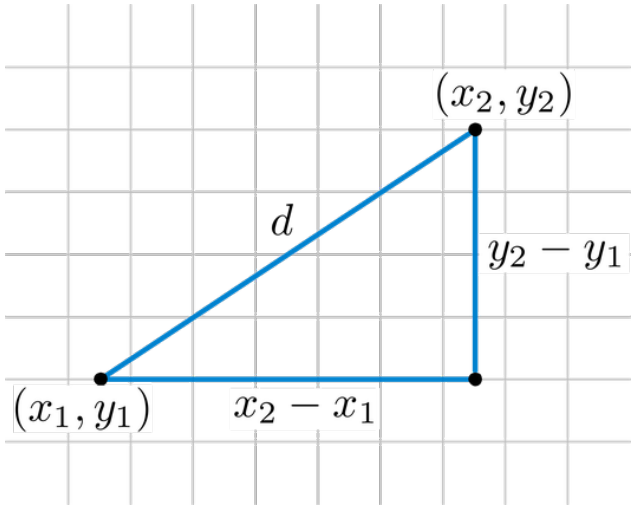
Standard norms

- The most well-known is **Euclidean norm:**

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$$

Quiz: Why is the modulo sign here?

A: x may be complex



Standard norms

- Euclidean norm is a special case of important p-norms:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

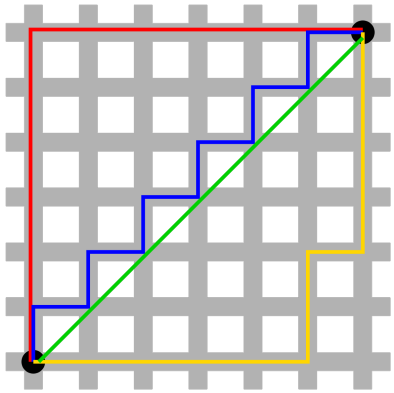
- Two important special cases:
  - **Infinity** norm, or Chebyshev norm is the maximum element: $\|x\|_\infty = max_i |x|_i$
  - L_1 norm, or Manhattan norm is the sum of modules: $\|x\|_1 = \sum_{i=1}^{n} \|x\|_i$

Standard norms

- Euclidean norm is a special case of important p-norms:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

- Two important special cases:
  - **Infinity** norm, or Chebyshev norm is the maximum element: $\|x\|_\infty = max_i |x|_i$
  - L_1 norm, or Manhattan norm is the sum of modules: $\|x\|_1 = \sum_{i=1}^{n} \|x\|_i$



Dot product

- Dot product (**scalar** product or **inner** product) of two vectors X and Y is the length of projection of X on the unit vector $\hat{Y}$:

$$X \cdot Y = \|X\| \, \|Y\| \cos\theta$$

- An easier definition:

$$X \cdot Y = \sum_{i=1}^{n} x_i y_i$$

- Useful properties:
  - Commutative: $X \cdot Y = Y \cdot X$
  - Distributive: $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
  - Zero iff X and Y are orthogonal

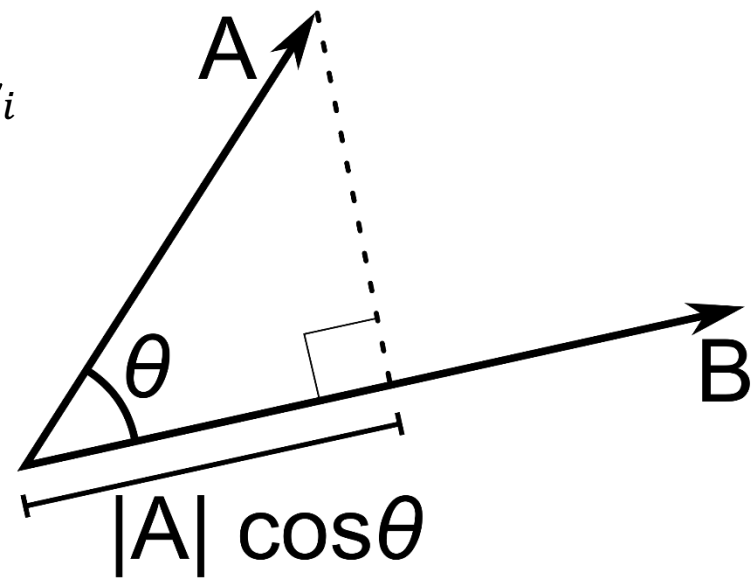
Dot product

- Dot product (**scalar** product or **inner** product) of two vectors X and Y is the length of projection of X on the unit vector $\hat{Y}$:

$$X \cdot Y = \|X\| \, \|Y\| \cos\theta$$

- An easier definition:

$$X \cdot Y = \sum_{i=1}^{n} x_i y_i$$



- Useful properties:
  - Commutative: $X \cdot Y = Y \cdot X$
  - Distributive: $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
  - Zero iff X and Y are orthogonal

Matrices

- It is *enough* to think a matrix to be a 2D array of numbers:

$$A = [a_{ij}], \qquad i = 1, \dots, n, \qquad j = 1, \dots, m$$

  - If $n = m$, we have a **square** matrix
  - If $n > m$, we have a **tall** matrix
  - If $n < m$, we have a **long** matrix
- A matrix takes nm memory to store
- If a matrix have (mostly) zeros, it is called **sparse**, otherwise it is said to be **dense**
- $A^T = [a_{ji}]$ is called matrix **transpose**

Matrix norms

- $\|A\|$ is called a **matrix norm** if:
  - $\|A\| \geq 0$
  - $\|A\| = 0$ iff $A = 0$
  - $\|\alpha A\| = |\alpha| \cdot \|A\|$
  - $\|A + B\| \leq \|A\| + \|B\|$
- Frobenius norm $\|A\|_F$ is the most simple:

$$\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |a_{ij}|^2}$$

Matrix norms

- Important class of norms are **operator norms**:

$$\|A\|_{op} = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

- Matrix p-norms are operator norms:
  - $p = 2$, **spectral** norm, denoted by $\|A\|_2$
  - $p = \infty$, $\|A\|_\infty = \max_i \sum_j |a_{ij}|$
  - $p = 1$, $\|A\|_1 = \max_j \sum_i |a_{ij}|$
- Spectral norm **can not** be computed directly from A entries!
- $\|A\|_2$ is the *largest singular value* of the matrix A, $\sigma_1(A)$

Later in the slides

Matrix-vector product

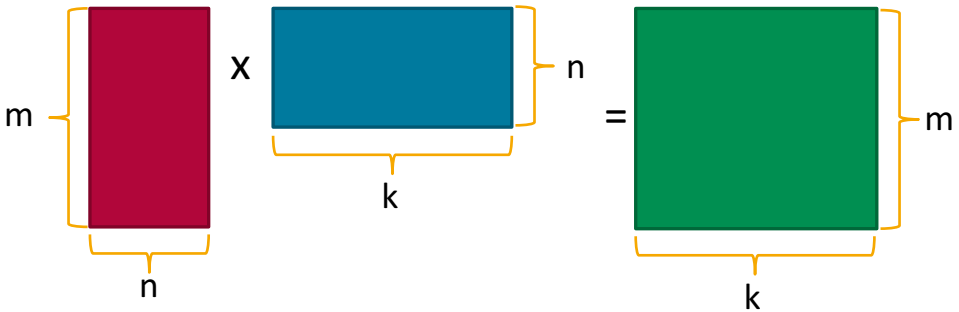
- Multiplication of $n \times n$ matrix A and $n \times 1$ vector x is defined as:

$$y_i = \sum_{j=1}^{n} a_{ij} x_j$$

- Overall complexity is $O(n^2)$
- For sparse matrices, it is only $O(\mathrm{nnz})$

Number of non-zero elements

Matrix multiplication



- Matrix multiplication is a basic building block of many algorithms, including graph mining
- A product of a $n \times k$ matrix A and $k \times m$ matrix B is defined as $n \times m$ matrix C:

$$c_{ij} = \sum_{s=1}^{k} a_{is} b_{sj}\,, \qquad i = 1, \dots, n, \qquad j = 1, \dots, m$$

- Time complexity is $O(nmk)$, $O(n^3)$ if matrices are square
- More efficient algorithms for large square matrices, $O(n^{2.8})$
- Efficient implementation is crucial (LAPACK, ATLAS, MKL, …)

Time complexity refresher

- $\sim 3 * 10^8$ users on Twitter, $\sim 1.1 * 10^{10}$ edges
- $\sim 2 * 10^9$ users on Facebook, $\sim 1.5 * 10^{11}$ edges
- Top i9 processor can do $\sim 1 * 10^{12}$ FLOPS

	Sparse matvec	**Dense matvec**	**Dense matmul**
Twitter	$\frac{1.1 * 10^{10}}{10^{12}} \approx 10^{-2} s$	$\frac{(3 * 10^8)^2}{10^{12}} \approx 1\ day$	$\frac{(3 * 10^8)^{2.8}}{10^{12}} \approx 20000\ years$
Facebook	$\frac{1.5 * 10^{11}}{10^{12}} \approx 10^{-1} s$	$\frac{(2 * 10^9)^2}{10^{12}} \approx 45\ days$	$\frac{(2 * 10^9)^{2.8}}{10^{12}} \approx 4\ million\ years$

Linear dependence

- Matrix can be considered to be a sequence of columns:

$$A = [a_1, a_2, \dots, a_m]$$

- Vectors a_i are called linearly dependent if there exists non-zero coefficients x_i such that:

$$\sum_i a_i x_i = 0$$

- **Matrix rank** is the number of linearly independent columns
  - In real world, it is reasonable to assume low rank of a matrix
  - Rank of a random $n \times n$ matrix $\mathrm{A} \sim N(0,1)$ is n

Distribution symbol

Matrix rank

- **Matrix rank** is the number of linearly independent columns
- In real world, it is reasonable to assume low ran of a matrix
- Rank of a random $n \times n$ matrix $\mathrm{A} \sim N(0,1)$ is n
- Matrix rank is **unstable**:

$$\forall A \in \mathbb{R}^{n \times n}\ \exists B: rank(B) = n \wedge \|A - B\| = \epsilon$$

- Q: Does it mean that rank does not make sense numerically?
- A: **No**. We want to find B: $\|A - B\| = \epsilon$ and $\boldsymbol{B}$ **has minimal rank**

Dimensionality reduction

Johnson-Lindenstrauss lemma

Let $\mathrm{N} \gg 1$. Given $0 < \epsilon < 1$, a set of m points in $\mathbb{R}^N$, $n < \frac{8\log(m)}{\epsilon^2}$, and linear map f from $\mathbb{R}^N \to \mathbb{R}^n$:

$$(1-\epsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2$$

- Not very practical due to dependence on ϵ
- This lemma does not give a recipe how to construct f

Eigenvalues and eigenvectors

- Vector $s \neq 0$ is an **eigenvector** of square matrix A if λ exists:

$$As = s$$

- λ is an **eigenvalue** corresponding to s
- If a $n \times n$ matrix A has n eigenvectors $s_i, i = 1, \dots, n$, then:

$$AS = S\Lambda, \text{ where } S = (s_1, \dots, s_n), \Lambda = diag(\lambda_1, \dots, \lambda_n)$$

is called eigendecomposition of matrix A

Not all matrices are diagonalizable, but undirected graphs are

Singular Value Decomposition

Theorem. Any $n \times m$ matrix A can be written as a product:

$$A = U \Sigma V^T$$

where U is $n \times K$ and V is $m \times K$ matrix, $K = \min(m, n)$,

Σ is a diagonal matrix with elements $\sigma_1 \geq \cdots \geq \sigma_K$

If $\mathrm{rank}(A) = r$, then $\sigma_{r+1} = \cdots = \sigma_K = 0$.

Singular Value Decomposition

Eckart-Young theorem

The best low-rank approximation can be computed by SVD

More formally, let $\mathrm{r} < \mathrm{rank}(A)$, $A_r = U_r \Sigma_r V_r^T$. Then,

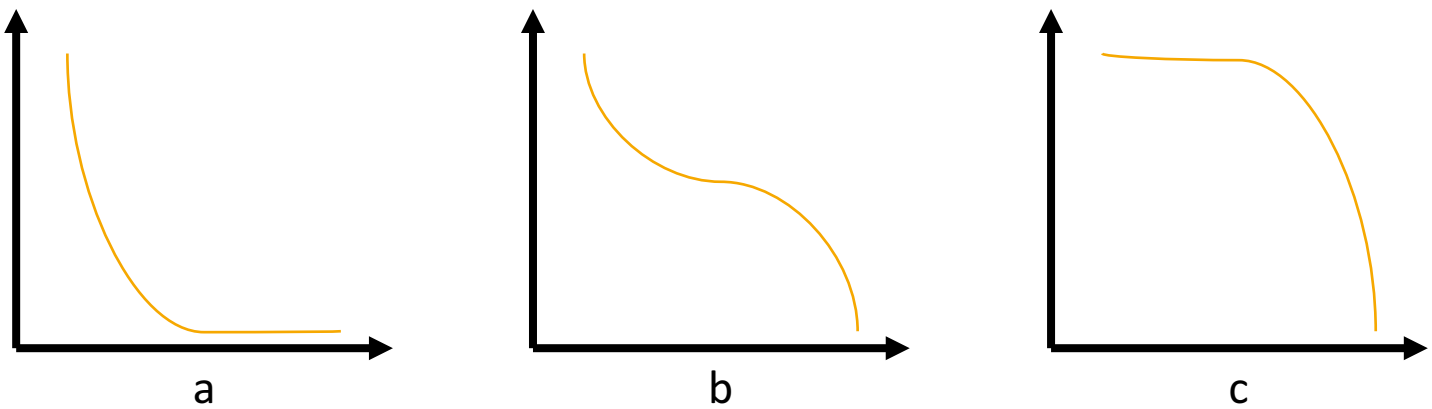
$$\min_{rank(B)=r} \|A - B\|_2 = \|A - A_r\|_2 = \sigma_{r+1}$$

The same holds for $\|\cdot\|_F$, and $\|A - A_r\|_F = \sqrt{\sigma_{r+1}^2 + \cdots + \sigma_K^2}$

Singular Value Decomposition

Singular value decay

Quiz: which is better?




ANY
QUESTIONS
?