# Lecture 5: Principal Component Analysis and SVD

## Foundations of Data Science: Algorithms and Mathematical Foundations

Mihai Cucuringu
mihai.cucuringu@stats.ox.ac.uk

CDT in Mathematics of Random System
University of Oxford

21 September, 2023

Singular Value Decomposition (SVD)

Principal Component Analysis

Computational considerations

Robust PCA

# Singular Value Decomposition (SVD)

Given $M \in \mathbb{R}^{m \times n}$, factorization into a product of three matrices:

$$M = U \Sigma V^T,$$

where

- $U \in O(m)$ i.e. $UU^T = U^T U = I$ (its columns are the left singular vectors)
- $V \in O(n)$ (its columns are the right singular vectors)
- $\Sigma$ is a diagonal matrix, with $\Sigma_{ii} \neq 0,$ for $i = 1, \ldots, min(m, n)$ being the singular values of $M$
- SVD defined for all matrices (rectangular or square) unlike the popular spectral decomposition

# Singular Value Decomposition (SVD)

- in many applications, the data matrix $M$ is close to a matrix of low-rank, and the goal is to find a low-rank matrix which is a good approximation to the data matrix
- consider the SVD of $M$ (sum of weighted rank-1 matrices)

$$M = \sum_{i=1}^{r} \sigma_i u_i v_i^T \tag{1}$$

- for $k = 1, 2, \ldots, r$, consider the sum truncated after $k$ terms

$$M_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T \tag{2}$$

  which renders $M_k$ to be of rank $k$
- one can show that $M_k$ is the best rank-$k$ approximation to $M$, where error is measured in Frobenius norm (Eckart–Young–Mirsky theorem)
- recall that for any matrix $M$, the sum of squares of its singular values equals the square of the Frobenius norm

$$\sum_{i=1} \sigma_i^2(M) = ||M||_F^2 := \left( \sum_{i=1}^{m} \sum_{j=1}^{n} M_{ij}^2 \right)^2 \tag{3}$$

# SVD - Computational Considerations

- $M_{n \times n}$ is often a very sparse matrix: each entry exists with probability $p$ (sampling probability)
- the leading singular values and singular vectors can be computed using iterative techniques at a typical cost of $O(pn^2)$
- eg., via a simple *power method*: all iterations only require a matrix-vector product at a cost of $pn^2$
- in the sparse setting, the computational cost is essentially linear in the number of nonzero entries (m) in the matrix (eg, nonzero entries in the adjacency matrix (edges in the graph))

# Spectral Decomposition

- If $M$ is an $n \times n$ symmetric matrix, then

$$M = V \Lambda V^T = \sum_{k=1}^{n} \lambda_k v_k v_k^T,$$

where

  - columns of $V$ are eigenvectors and
  - $\Lambda_{ii}$ is a diagonal matrix with entries given by eigenvalues $\lambda_i$ of $M$

- A real symmetric matrix $M_{n \times n}$ is positive semi-definite (PSD), denoted $M \succcurlyeq 0$, if $z^T M z \geq 0$ for all $z \in \mathbb{R}^d$
- Let $M$ be a real symmetric $n \times n$ matrix. Then, $M$ is positive semi-definite iff all its eigenvalues $\lambda_i \geq 0$.
- Spectral norm of matrix: $\|M\|_2 = |\lambda_{max}(M)|$

# Remarks SVD/eigen-decomposition

- the $u_i$ are eigenvectors of $MM^T$ and the $v_i$ are eigenvectors of $M^TM$
- $MM^T$ and $M^TM$ are positive semidefinite, so their eigenvalues are nonnegative
- if $\lambda_i$ are the eigenvalues of $M^TM$, then $\sigma_i^2 = \lambda_i$ if $\lambda_i > 0$
- if $M$ is square and Hermitian, then the SVD and the eigenvalue decomposition are the same

# Some properties

- $Tr(M) = \sum_{k=1}^{n} M_{kk} = \sum_{k=1}^{n} \lambda_k(M)$
- Frobenius norm, $\|M\|_F = \sqrt{\sum_{i,j} M_{ij}^2} = \sqrt{Tr(M^T M)}$
- $\mathrm{Tr}(AB) = \sum_{i,j=1}^{n} A_{ij} B_{ji} = \mathrm{Tr}(BA)$
- trace is invariant under cyclic permutation

$$\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA) = \mathrm{Tr}(CAB) \neq \mathrm{Tr}(ACB)$$

# Quadratic Forms

- For $M$ a symmetric matrix, interested in solving

$$\max_{\substack{V \in \mathbb{R}^{n \times d} \\ V^T V = I_{d \times d}}} Tr(V^T M V) \iff \max_{\substack{v_1, \ldots, v_n \in \mathbb{R}^d \\ v_i^T v_j = \delta_{ij}}} \sum_{k=1}^{n} v_k^T M v_k$$

where $\delta_{ij}$ is the Kronecker delta

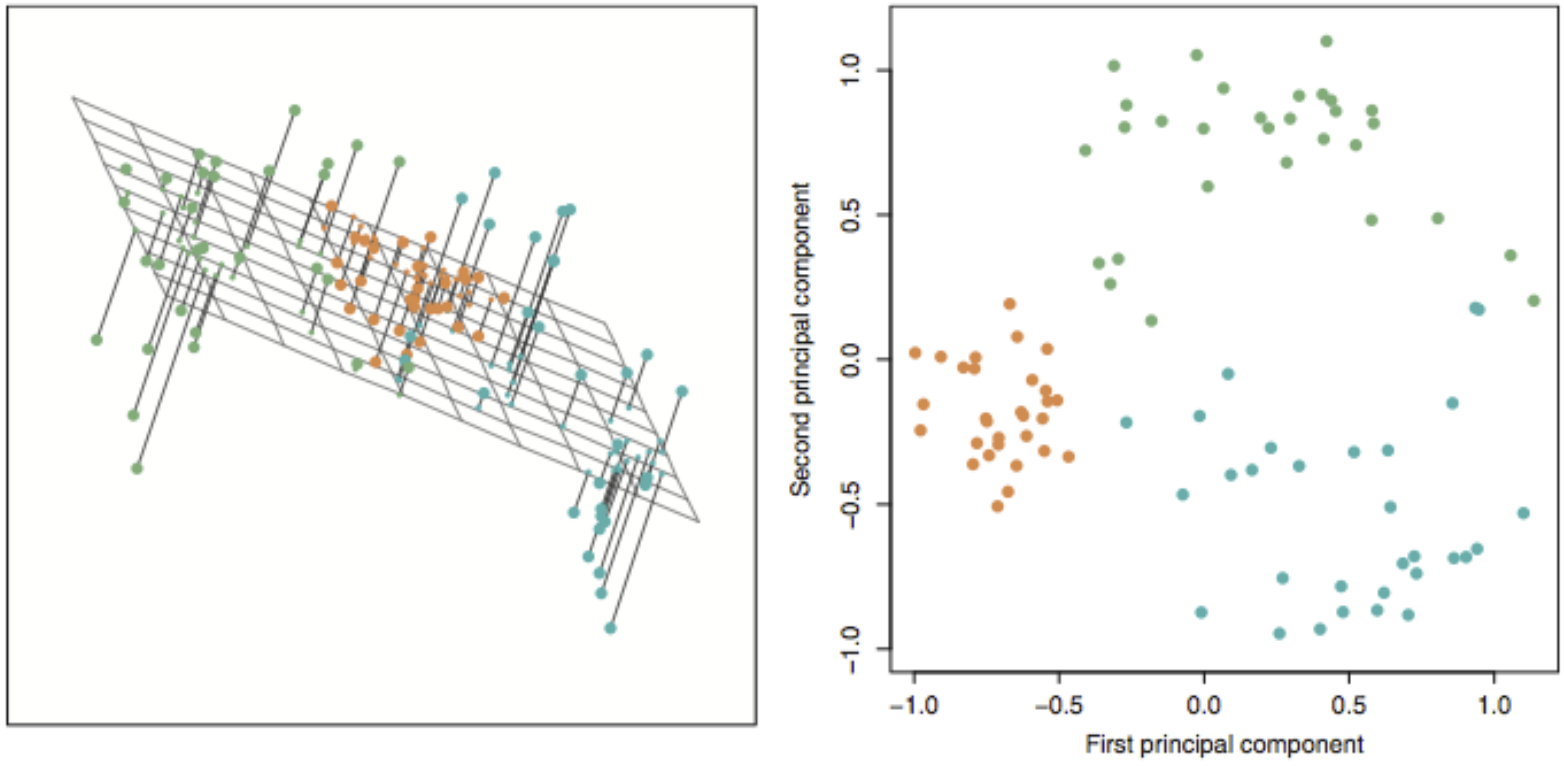
$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

- If $d = 1$, this amounts to

$$\max_{\substack{v \in \mathbb{R}^n \\ ||v||_2 = 1}} v^T M v = \lambda_{max}(M)$$

which is maximized by $v$, the leading eigenvector of $M$, corresponding to the largest eigenvalue.

# PCA: example, $n = 90$, $p = 3$, $d = 2$




- Left: top two PC directions that span the plane that best fits the data.
  - minimizes the sum of distances$^2$ from points to the plane
- Right: top two PC score vectors giving the coordinates of the projection of the 90 observations onto the plane
  - The variance in the plane is maximized

PCA dates back to a 1901 paper by Karl Pearson!

# Principal Component Analysis - Theoretical considerations

*Dimensionality reduction:* Given $x_1, \ldots, x_n \in \mathbb{R}^p$ for large $p$, the goal is to find a lower-dimension representation of the data, and transform data into $n \times d$ where $d \ll p$ by a linear projection.

- PCA is a linear technique: fits the best hyperplane through data points by projecting the $n$ points onto a $d$-dim space

Two equivalent formulations that lead to the same solution:

1. Find the best possible affine $d$-dim space that fits the given data (minimize residual). Approx. each $x_k$, $k = 1, \ldots, n$

$$\overbrace{x_k}^{p\times 1 \text{ approximation}} \approx \underbrace{\mu}_{\text{const translation vector}} + \underbrace{\sum_{i=1}^{d} (\beta_k)_i v_i}_{\text{lin comb of basis of } d\text{-dim space}}$$

$$\underbrace{x_k}_{p\times 1} \approx \underbrace{\mu}_{p\times 1} + \underbrace{V}_{p\times d}\underbrace{\beta_k}_{d\times 1} \quad \text{where} \quad V^T V = I_{d\times d}$$

2. Find d-dimensional projection of $x_1, \ldots, x_n$, in order to maximize the variance of the projected points.

# PCA Approach I - best affine $d$-dim space optimization formulation

[following book by A. Bandeira, A. Singer and T. Strohmer]

Notation

- Sample mean

$$\mu_n = \frac{1}{n}\sum_{k=1}^{n} x_k$$

- Sample covariance matrix

$$\Sigma_M = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \mu_n)(x_k - \mu_n)^T$$

# Measuring goodness of fit

- In the ”least-squares” sense, the PCA objective is:

$$\min_{\substack{\mu, V, \beta_k \\ V^T V = I_{dxd}}} \quad \sum_{k=1}^{n} ||x_k - (\mu + V \cdot \beta_k)||_2^2$$

- WLOG, we set $\sum \beta_k = 0$, i.e., ”center” the points around the origin.

# Optimizing for $\mu$

First-order conditions for $\mu$ correspond to

$$\nabla_\mu \left( \sum_{k=1}^{n} ||x_k - \mu - V \cdot \beta_k||_2^2 \right) = 0$$

$$\iff \sum_{k=1}^{n} (x_k - \mu - V \cdot \beta_k) = 0$$

$$\iff \sum_{k=1}^{n} x_k - n\mu - V \sum_{k=1}^{n} \beta_k = 0$$

Recalling $\sum_{k=1}^{n} \beta_k = 0$ from earlier, yields

$$\boxed{\mu^* = \frac{1}{n} \sum_{k=1}^{n} x_k}$$

which is the sample mean $\mu_n$ previously defined.

# Optimizing for $\beta_k$

- We now have

$$\mu^* = \frac{1}{n}\sum_{k=1}^{n} x_k$$

- Recall that V $\in \mathbb{R}^{p\times d}$
- The optimization problem amounts to

$$\begin{aligned} \underset{V,\beta_k}{\text{minimize}} \quad & \sum_{k=1}^{n} ||x_k - \mu_n - V \cdot \beta_k||_2^2 \\ \text{subject to} \quad & V^T \cdot V = I_{dxd} \end{aligned}$$

- we can rewrite the objective function as

$$||x_1 - \mu_n - V \cdot \beta_1||_2^2 + ||x_2 - \mu_n - V \cdot \beta_2||_2^2 + \cdots$$

- the problem **decouples** for each $k$, so we can focus on individual terms

# Optimizing for $\beta_k$ (cont.)

For a given $k$, our optimization problem amounts to:

$$\min_{\beta_k} ||x_k - \mu_n - V \cdot \beta_k||_2^2$$

$$= \min_{\beta_k} ||x_k - \mu_n - \sum_{i=1}^{d} v_i \cdot \beta_{k_i}||_2^2$$

$$= \nabla_{(\beta_k)_i}(||x_k - \mu_n - v_i(\beta_k)_1 - ... - v_i(\beta_k)_i - ... - v_d(\beta_k)_d||_2^2)$$

$$\iff v_i^T[x_k - \mu_n - V \cdot \beta_k]^1 = 0$$

# Optimizing for $\beta_k$ (cont.)

Since

$$V \cdot \beta_k = v_1(\beta_k)_1 + ... + v_d(\beta_k)_d,$$

and

$$v_i^T \cdot v_j = 0 \text{ for } i \neq j,$$

and

$$v_i^T \cdot v_j = 1 \text{ for } i = j,$$

we arrive at

$$v_i^T(x_k - \mu_n) - v_i^T \cdot V \cdot \beta_k = 0$$

$$(\beta_k^*)_i = v_i^T(x_k - \mu_n), \forall i = 1, \ldots, d$$

In vector form:

$$\boxed{(\beta_k^*) = V^T(x_k - \mu_n)}$$

# Optimizing for $V$

- we have $\beta_k^* = V^T(x_k - \mu_n)$
- substituting $\beta_k^*$ into the optimization problem yields

$$\begin{aligned} \underset{V}{\text{minimize}} \quad & \sum_{k=1}^{n} ||(x_k - \mu_n) - V \cdot V^T(x_k - \mu_n)||_2^2 \\ \text{subject to} \quad & V^T \cdot V = I_{dxd} \end{aligned}$$

- if we consider for each individual term, we have:

$$\begin{aligned} & ||x_k - \mu_n - V \cdot V^T(x_k - \mu_n)||_2^2 \\ & = [x_k^T - \mu_n^T - (x_k - \mu_n)^T \cdot V \cdot V^T][x_k - \mu_n - V \cdot V^T(x_k - \mu_n)]d \end{aligned}$$

- keep in mind that $V^T \cdot V = I$

# PCA derivation - recap

We have reduced the original problem

$$\underset{\substack{\mu, V, \beta_k \\ V^T V = \mathbb{I}_{d\times d}}}{\text{minimize}} \sum_{k=1}^{n} \|x_k - \mu - V\beta_k\|_2^2 \tag{4}$$

to

$$\underset{V^T V = \mathbb{I}_{d\times d}}{\text{minimize}} \sum_{k=1}^{n} \left\| x_k - \mu_n - VV^\top (x_k - \mu_n) \right\|_2^2 \tag{5}$$

Now, since $\|y\|_2^2 = y^\top y$, we have

$$\left\|x_k - \mu_n - VV^\top(x_k - \mu_n)\right\|_2^2 =$$

$$= (x_k - \mu_n - VV^\top(x_k - \mu_n))^\top(x_k - \mu_n - VV^\top(x_k - \mu_n)) \tag{6}$$

$$= (x_k - \mu_n)^\top(x_k - \mu_n) + (x_k - \mu_n)^\top \underbrace{VV^\top VV^\top}_{\mathbb{I}}(x_k - \mu_n)$$

$$- 2(x_k - \mu_n)^\top VV^\top(x_k - \mu_n) \tag{7}$$

$$= \underbrace{(x_k - \mu_n)^\top(x_k - \mu_n)}_{\text{does not depend on } V} - (x_k - \mu_n)^\top VV^\top(x_k - \mu_n) \tag{8}$$

Thus, (5) is equivalent to

$$\max_{V^\top V = \mathbb{I}_{d\times d}} \sum_{k=1}^{n} (x_k - \mu_n)^\top VV^\top(x_k - \mu_n) \tag{9}$$

By using properties of the trace, we see that

$$\sum_{k=1}^{n} \underbrace{(x_k - \mu_n)^\top}_{1 \times p} \underbrace{VV^\top}_{p \times p} \underbrace{(x_k - \mu_n)}_{p \times 1} =$$

$$= \sum_{k=1}^{n} \mathrm{Tr}\left[(x_k - \mu_n)^\top VV^\top (x_k - \mu_n)\right] \tag{10}$$

$$= \sum_{k=1}^{n} \mathrm{Tr}\left[V^\top (x_k - \mu_n)(x_k - \mu_n)^\top V\right] \tag{11}$$

$$= \mathrm{Tr}\left[\sum_{k=1}^{n} V^\top (x_k - \mu_n)(x_k - \mu_n)^\top V\right] \tag{12}$$

$$= \mathrm{Tr}\left[V^\top \sum_{k=1}^{n} (x_k - \mu_n)(x_k - \mu_n)^\top V\right] \tag{13}$$

$$= (n-1)\,\mathrm{Tr}(V^T \Sigma_n V) \tag{14}$$

where $\Sigma_n$ is the sample covariance matrix (of size $n \times n$).

# Finally ...

Thus, the solution to (9) is given by

$$\max_{V^\top V = \mathbb{I}_{d\times d}} \mathrm{Tr}(V^\top \Sigma_n V), \tag{15}$$

which is equivalent to

$$\max_{v_1 \ldots . v_d \in \mathbb{R}^n, v_i v_j = \delta_{ij}} \sum_{k=1}^{d} v_k^\top \Sigma_n v_k, \tag{16}$$

whose solution is given by the top $d$ leading eigenvectors of $\Sigma_n$.

# Derivation of PCA: Approach II - $d$-dimensional projection that preserves the most variance

Goal: We wish to

- find an orthonormal basis $\{v_1, \ldots, v_d\}$ of a $d$-dimensional subspace
- such that the projection of the original data $\{x_1, \ldots, x_n\}$ on this subspace has the most variance.

Mathematically:

- Let $V$ be a matrix of dimension $p \times d$ with $v_i$ $(i = 1, \ldots, d)$ as its $i^{th}$ column, so that $V^T V = \mathbb{I}_{d \times d}$.

- The projected points $y_k, (k = 1, \ldots, n)$ are given by

$$y_k = V^\top x_k$$

# PCA II - Optimize variance

- Goal: aim for the projected points $y_1, \ldots, y_n$ to have as much variance as possible
- Recall that $\mathrm{Var}[y] = \frac{1}{n-1} \sum_{k=1}^{n} (y_k - \bar{y})^2$; our goal is equivalent to

$$\max_{V^\top V = \mathbb{I}_{d\times d}} \sum_{k=1}^{n} \left\| V^\top x_k - \frac{1}{n} \sum_{i=1}^{n} V^\top x_i \right\|^2 \tag{17}$$

$$\iff \max_{V^\top V = \mathbb{I}_{d\times d}} \sum_{k=1}^{n} \left\| \underbrace{V^\top}_{d \times p} \underbrace{(x_k - \mu_n)}_{p\times 1} \right\|^2 \tag{18}$$

$$\iff \max_{V^\top V = \mathbb{I}_{d\times d}} \mathrm{Tr}(V^\top \Sigma_n V), \tag{19}$$

- since $\|A\|_F^2 = \mathrm{Tr}(A^\top A)$
- sample covariance matrix $\Sigma = \dfrac{1}{n-1} \sum_{k=1}^{n} (x_k - \mu_n)(x_k - \mu_n)^T$
- The two approaches for the derivation of PCA are equivalent.

## First direction in which variance is maximized ($d = 1$)

Perform PCA on $X$, an $n \times p$ matrix with rows $x_i, i = 1, \ldots, n$

Let $B = \{v_1, v_2, ..., v_d\}$ be an orthonormal basis of the $d-$dimensional subspace, and let

$$v_1 = \underset{v_i \in B,\ v_i^T v_i = 1}{\text{argmax}} \ \text{Var}[Xv_i]$$

Rewrite this in terms of the covariance matrix $C = \frac{1}{n-1} X^\top X$

$$\begin{aligned} \text{Var}[Xv_i] &= \tfrac{1}{n-1}(Xv_i)^\top (Xv_i) && (20) \\ &= v_i \tfrac{1}{n-1} X^\top X v_i && (21) \\ &= v_i^\top C v_i && (22) \end{aligned}$$

Solve this constrained optimization by using Lagrange Multipliers

$$L(v_1, \lambda_1) = v_1^\top C v_1 + \lambda_1 (1 - v_1^\top v_1) \quad (23)$$

By imposing the condition $\nabla_{v_1} L(v_1, \lambda_1) = 0$, we have

$$2Cv_1 - 2\lambda_1 v_1 = 0 \Rightarrow Cv_1 = \lambda_1 v_1 \Rightarrow \lambda_1 = v_1^\top C v_1$$

i.e. the projection variance is the (first) eigenvalue.

# Intuition

- PCA considers the eigenvector decomposition of $\Sigma_n$
- analyzes the projection of the centered data points (after having subtracted the sample mean $\mu_n$) on the top $k$ eigenvectors of the sample covariance matrix $\Sigma_n$ as *principal components*
- where the top $k$ eigenvectors are those associated with the largest $k$ eigenvalues

What about the cost?

# Computing the Principal Components (I - via spectral decomposition)

Algorithmic considerations:

- Compute top eigenvectors of the sample covariance matrix $\Sigma_n$ of size $p \times p$

$$\Sigma_n = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \mu)(x_k - \mu)^T$$

where $\mu$ is the sample mean vector.

Naive way

- building $\Sigma_n$ takes $O(np^2)$
- finding the spectral decomposition takes $O(p^3)$ work
- overall complexity $O(\max\{np^2, p^3\})$

# Computing the Principal Components (II - via SVD)

Let $X$ of size $p \times n$ be given by $X = [x_1, x_2, \ldots, x_n]$

$$\Sigma_n = \frac{1}{n}\sum_{k=1}^{n}(x_k - \mu)(x_k - \mu)^T = \frac{1}{n}\left(X - \mu \mathbf{1}^T\right)\left(X - \mu \mathbf{1}^T\right)^T \tag{24}$$

Consider the SVD of

$$X - \mu \mathbf{1}^T = LDR^T$$

then

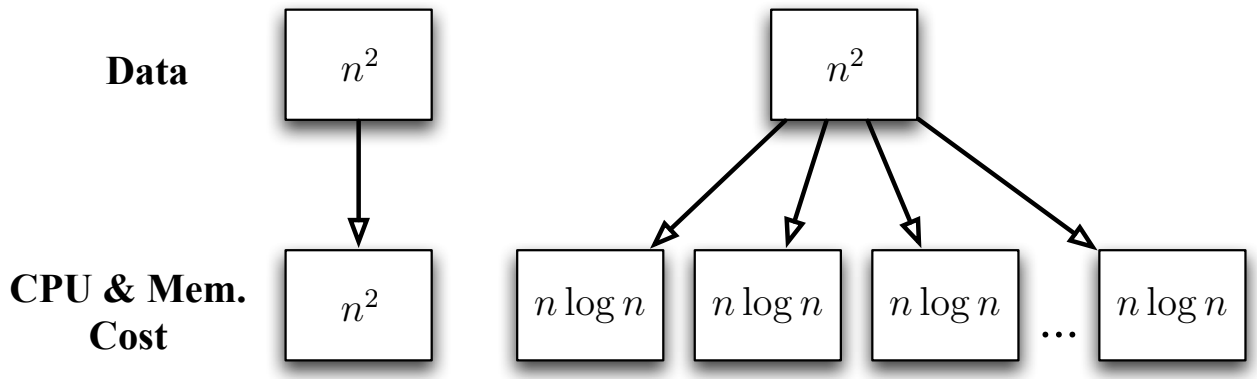
$$\Sigma_n = \frac{1}{n}(X - \mu \mathbf{1}^T)(X - \mu \mathbf{1}^T)^T = \frac{1}{n} LD \underbrace{R^T R}_{=\mathbb{I}} DL^T = \frac{1}{n} LD^2 L^T \tag{25}$$

so $L$ corresponds to the eigenvectors of $\Sigma_n$. Why bother?
Computing the SVD of $(X - \mu \mathbf{1}^T)$ takes $O(\min(n^2 p, p^2 n))$

- if interested only in the top $d$ eigenvectors $\Rightarrow O(npd)$
- randomized algorithms for an approximate solution $\Rightarrow O(pn \log d + (p+n)d^2)$

# Distributed SVD - decentralised implementation

- distributed computation of the extremal eigenvectors
- El Karoui, N. and d'Aspremont, A., 2010. *Second order accurate distributed eigenvector computation for extremely large matrices*. Electronic Journal of Statistics, 4, pp.1345-1385
- show that averaging eigenvectors of randomly subsampled matrices efficiently approximates the true eigenvectors of the original matrix, under certain conditions on the spectral decomposition



# Distributed SVD - Matrix Subsampling

Subsampling procedure in

- [AM07] Achlioptas & McSherry. *Fast computation of low-rank matrix approximations*. Journal of the ACM, 54(2), 2007.

approximates a symmetric matrix $M$ using a subset of its coefficients.

- the entries of $M$ are independently sampled as

$$S_{ij} = \begin{cases} M_{ij}/p & \text{with probability } p \\ 0 & \text{otherwise} \end{cases}$$

  where $p \in [0, 1]$ is the sampling probability

- Thm 1.4 in [AM07] shows that when $n$ is large enough

$$||M - S||_2 \leq 4||M||_\infty \sqrt{n/p} \tag{26}$$

  holds with high probability.

# Matrix Subsampling

- consider the spectral decomposition of $M = \sum_{i=1}^{n} \lambda_i u_i u_i^T$ with eigenvalues $\lambda_1 > \lambda_2 > \ldots \lambda_n$
- define subsampling matrix $Q \in S_n$ with idd Bernoulli coeffs

$$Q_{ij} = \begin{cases} 1/p & \text{with probability } p \\ 0 & \text{otherwise} \end{cases}$$

- leading to

$$Q = 11^T + \sqrt{\frac{1-p}{p}} C \tag{27}$$

where $C$ has idd entries with mean zero and variance once, as

$$C_{ij} = \begin{cases} \sqrt{(1-p)p} & \text{with probability } p \\ -\sqrt{p/(1-p)} & \text{otherwise} \end{cases}$$

- allowing to write the sampled matrix as

$$S = M \circ Q = M + \sqrt{\frac{1-p}{p}} \left( \sum_{i=1}^{n} \lambda_i (u_i u_i^T) \circ C \right) \equiv M + E \tag{28}$$

- bound the spectral norm of the residual matrix $E$ as $n \to \infty$; is $||E||_2$ is small, then $S$ is a good approx. of $M$ in spectral terms.

# Issues with PCA

- cannot handle nonlinearity in the data
- very often, the real data has samples grossly corrupted (measurement errors, adversarial attacks)
- sensitivity to outliers - PCA fails even with a few outliers

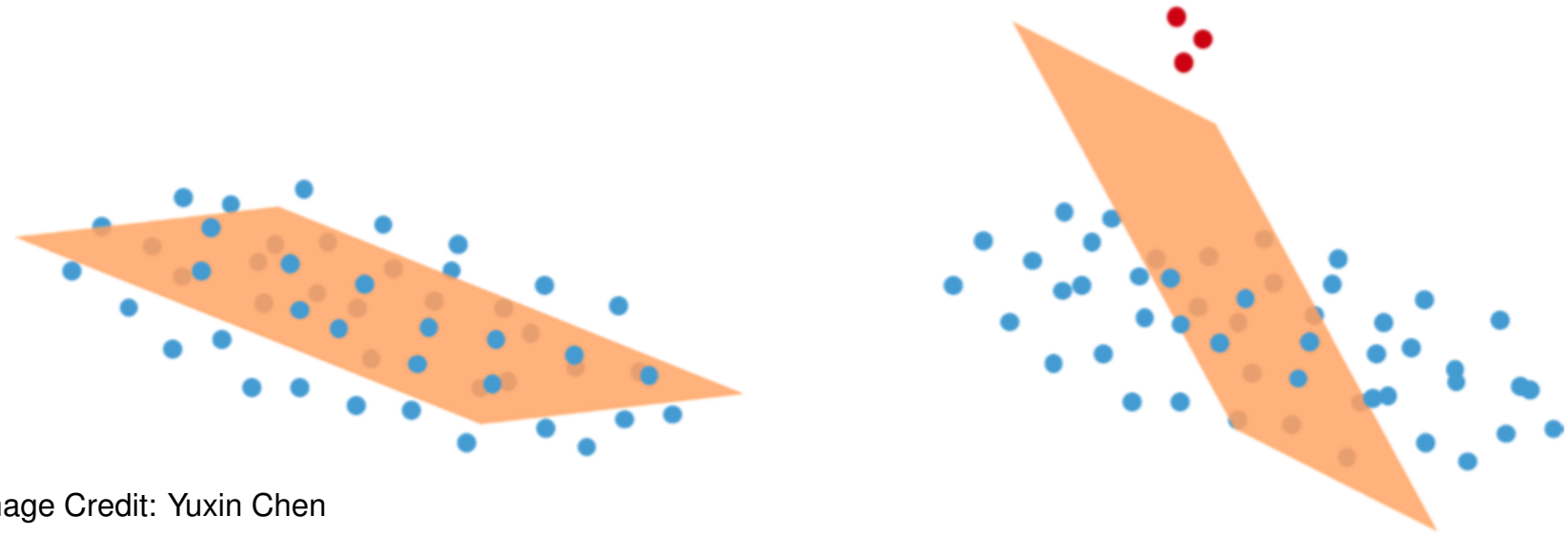


Image Credit: Yuxin Chen

- another way to look at PCA

$$\text{minimize}_{L:rank(L)=k} ||X - L||_F \tag{29}$$

where $X$ is the data matrix of size $p \times n$ with the $n$ samples $x_1, x_2, \ldots, x_n$

- find the best rank-k approximation of $X$

# Nuclear norm of a matrix

- rank: number of nonzero singular values of matrix $L_{n \times m}$
- rank minimization is NP-hard
- what to replace the rank constraint of $L$ with?
- the sum of the singular values of $L$, denoted as the **nuclear norm**

$$||L||_* := \sum_{i=1}^{\min\{n,m\}} \sigma_i(L) \tag{30}$$

  where $\sigma_i(L)$ denotes the i-th singular value of $L$
- the nuclear norm is the tightest convex relaxation of the rank constraint, i.e., the nuclear norm ball $\{L : ||L||_* \leq 1\}$ is the convex hull of the collection of unit-norm rank-1 matrices $\{uv^T : ||u||_2 = ||v||_2 = 1\}$
- can be shown to lead to a convex program that can be solved efficiently in polynomial time
- does not require knowledge of the rank a-priori

# Robust PCA - a convex relaxation

- ideally we would like to solve

$$\begin{aligned} \min_{L,S} \quad & \text{rank}(L) + \lambda ||S||_0 \\ \text{s.t.} \quad & M = L + S \end{aligned} \tag{31}$$

- relax to

$$\begin{aligned} \min_{L,S} \quad & ||L||_* + \lambda ||S||_1 \\ \text{s.t.} \quad & M = L + S \end{aligned} \tag{32}$$

- $||\cdot||_*$ is the *nuclear norm*
- $||\cdot||_1$ is the entry-wise $\ell_1$ norm
- $\lambda > 0$ is a regularization parameter balancing the two terms: the *low-rankness* vs the *sparsity*

(32) is exact with high probability, under the following conditions:

- for small enough rank (as high as $n/\text{polylog}(n)$)
- and randomly located nonzero entries in $S$
- and small enough $||S||_0 \leq cn^2$

Candès, Emmanuel J., Xiaodong Li, Yi Ma, and John Wright. *Robust principal component analysis?* Journal of the ACM (JACM) 58, no. 3 (2011): 1-37.
(Google scholar citations: 5430 (2020), 6380 (2021), 6580 (2022), 7500 (2023))