Dimensionality Reduction:
Singular Value Decomposition (SVD) &
Principal Component Analysis (PCA)

High-Dimensional Data

High-Dimensions = Lot of Features

- Document classification
  - Features per document = thousands of words/unigrams, millions of bigrams, contextual information

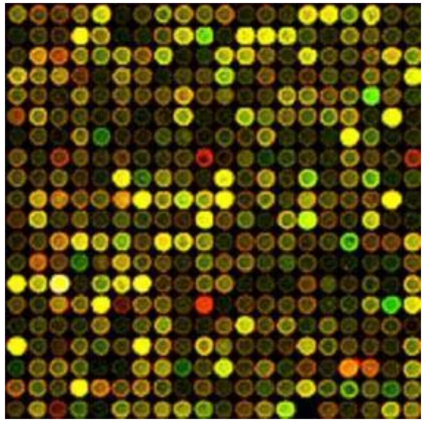
- Surveys – Netflix
  - 480189 users x 17770 movies

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

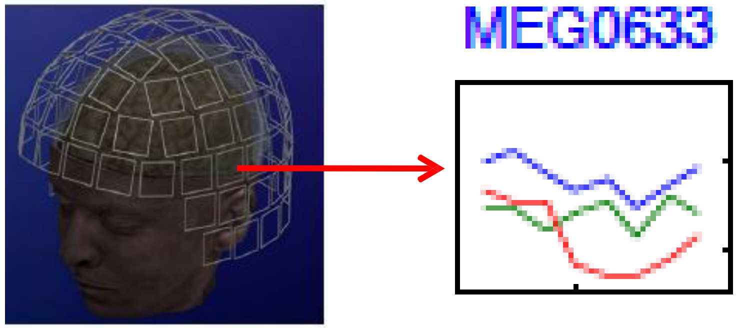
High-Dimensional Data

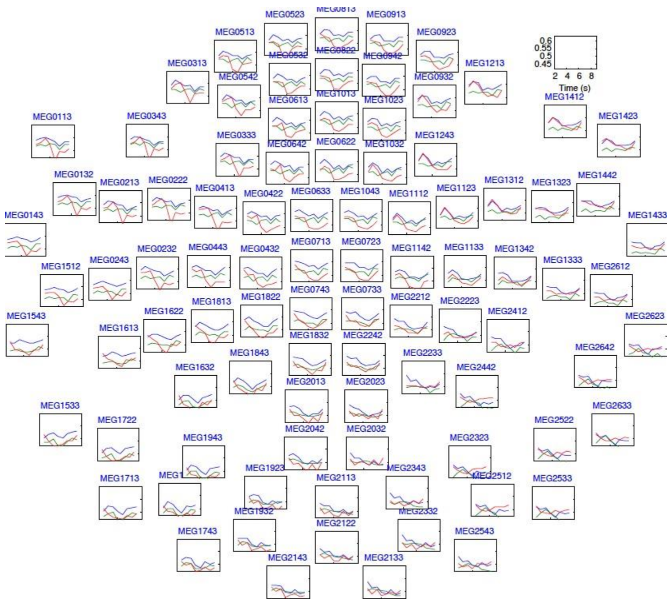
High-Dimensions = Lot of Features

- Discovering gene networks
  - 10,000 genes x 1000 drugs x several species



- MEG Brain Imaging
  - 120 locations x 500 time points x 20 objects





Curse of Dimensionality

Why are more features bad?

- Redundant features
  - not all words are useful to classify a document
- Hard to interpret and visualize
- Hard to store and process data
  - computationally challenging
- Complexity of decision boundaries tends to grow with # features

Dimensionality Reduction

Represent data with fewer dimensions

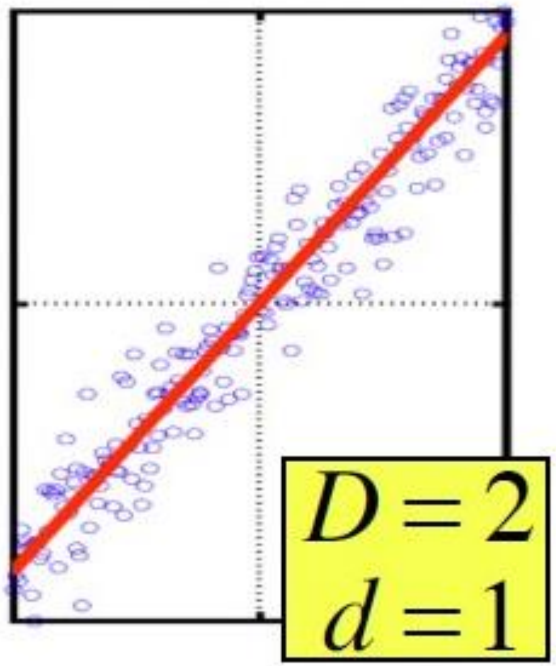
- **Easier learning** – fewer parameters
- **Visualization** – show high dimensional data in 2D
- Discover "**intrinsic dimensionality**" of data
  - high dimensional data that is truly lower dimensional
  - noise reduction

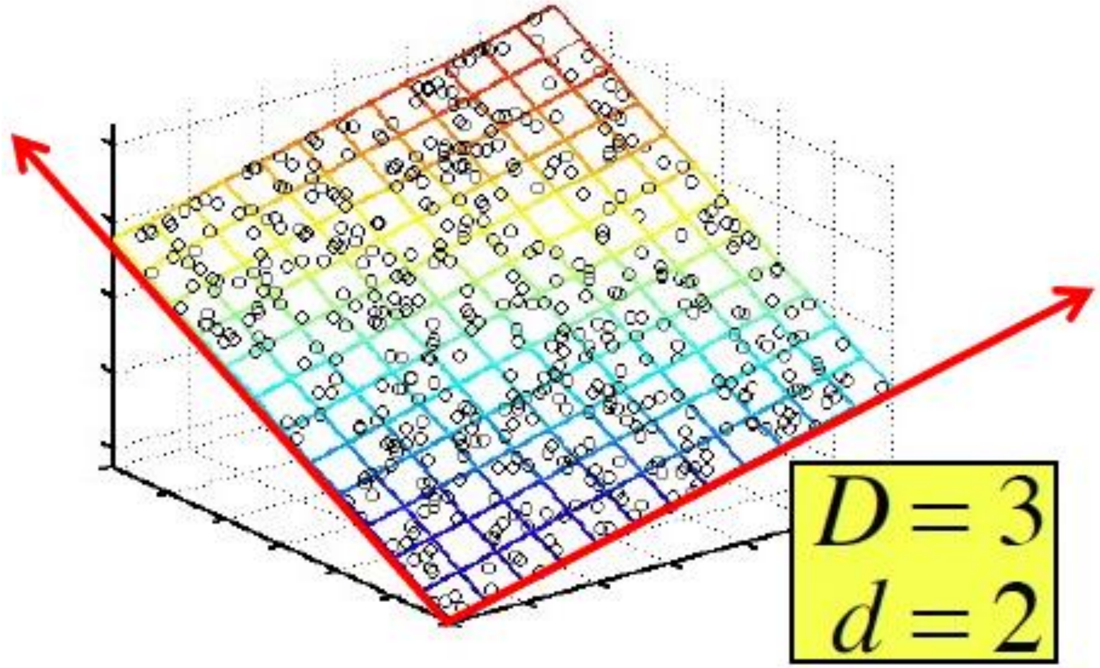
Informally: Given data points in *D*-dimensional space

- Convert them to data points in $d<D$ dimensions
- Goal: With minimal/maximal information loss/maintenance

Dimensionality Reduction

Assumption: Data (approximately) lies on a lower dimensional space





Singular Value Decomposition(SVD)

Orthonormal Basis

A set of vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{\boldsymbol{d}}\} \subset \mathbb{R}^d$ form an orthonormal basis if:

- unit ℓ_2-norm: $||\boldsymbol{u}_i||_2 = 1$, for all *i=1 to d*
- orthogonal to each other: $(\boldsymbol{u}_i, \boldsymbol{u}_j) = 0$, for all $i \neq j$

Let $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{\boldsymbol{d}}\}$ be any orthonormal basis of $\mathbb{R}^d$

- Any vector $\boldsymbol{x} \in \mathbb{R}^d$ can be expressed as a linear combination of $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{\boldsymbol{d}}$, for some real-valued $\alpha_1, \alpha_2, \ldots, \alpha_d$

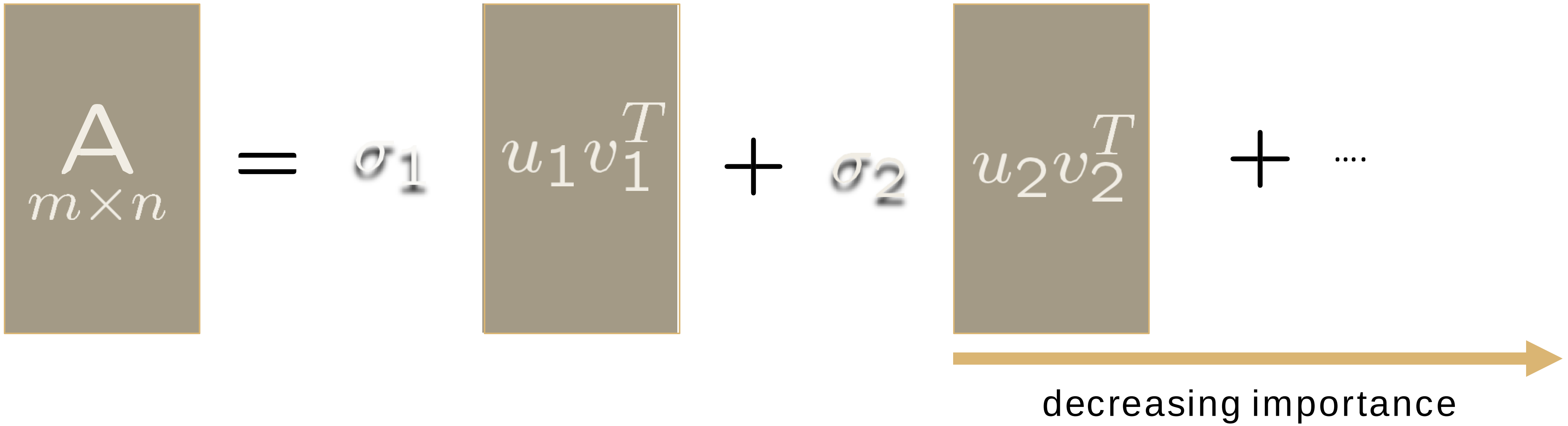
$$\mathbf{x} = \alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2 + \cdots + \alpha_d \mathbf{u}_d$$

Singular Value Decomposition (SVD)

- Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix
- Rank: r = rank($\mathbf{A}$) ($\leq m,n$)
- SVD: $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sum_{i=1}^{r} \sigma_i \, \boldsymbol{u}_i \boldsymbol{v}_i^T$
  - Singular values: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$
  - $\boldsymbol{\Sigma}$ diagonal matrix, with $\Sigma_{ii} = \sigma_i$
  - Left singular matrix/vectors: $\mathbf{U}=\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r\} \in \mathbb{R}^m$ forms an orthonormal basis, i.e., $\mathbf{U}^T \mathbf{U} = \mathbf{I}_r$
  - Right singular matrix/vectors: $\mathbf{V}=\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r\} \in \mathbb{R}^n$ forms an orthonormal basis, i.e., $\mathbf{V}^T \mathbf{V} = \mathbf{I}_r$

Singular Value Decomposition (SVD)

- Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix
- Rank: $r = \text{rank}(\mathbf{A})$ ($\le m, n$)
- SVD: $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$
  - Singular values: $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_r > 0$



Singular Value Decomposition (SVD)

- Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix
- Rank: r = rank($\mathbf{A}$) ($\leq m,n$)
- SVD: **A** = $\mathbf{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$
- Truncated SVD: for 0 < k < r
  - $\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$
  - $\mathbf{A}_k$ is the **best** rank-k approximation to $\mathbf{A}$

$$\mathbf{A}_k = \arg\min_{\mathbf{B}} \|\mathbf{A} - \mathbf{B}\|_F^2, \text{ s.t. } \operatorname{rank}(\mathbf{B}) \leq k.$$

Matrix Frobenius Norm

- Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix
- Frobenius norm

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

A generalization of the l_2-vector norm to matrix

- Relation with singular values

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2}$$

σ_i is the i-th singular value of $\mathbf{A}$

Error of Truncated SVD

- Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix and rank($\mathbf{A}$)=r
- SVD: $\mathbf{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$
- Truncated SVD: $\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ (for 0 < k < r)
- Error of the truncated SVD

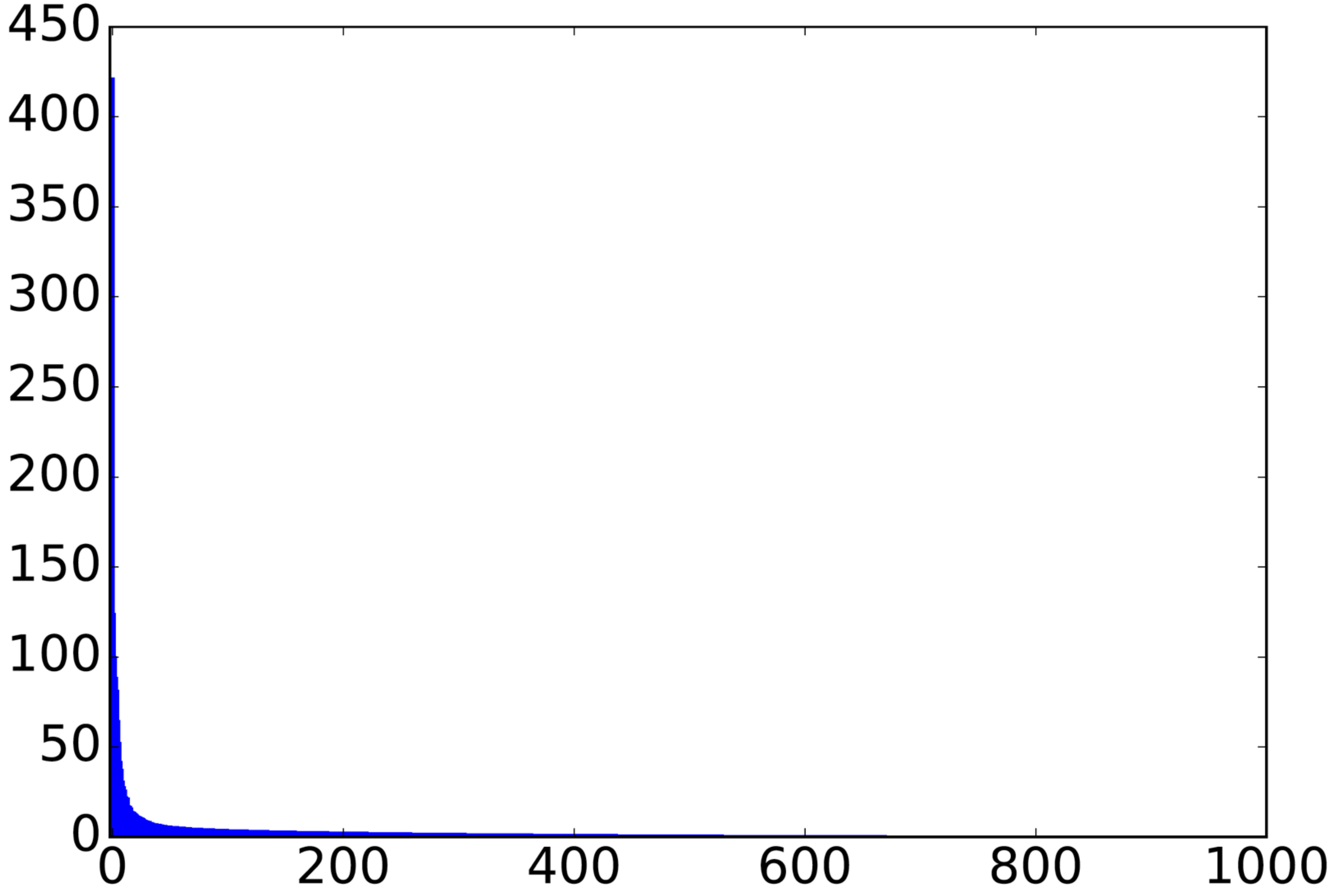
$$\|\mathbf{A} - \mathbf{A}_k\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2$$

Bottom (the smallest) singular values

SVD: Example

Original image size (1000 x 1500)

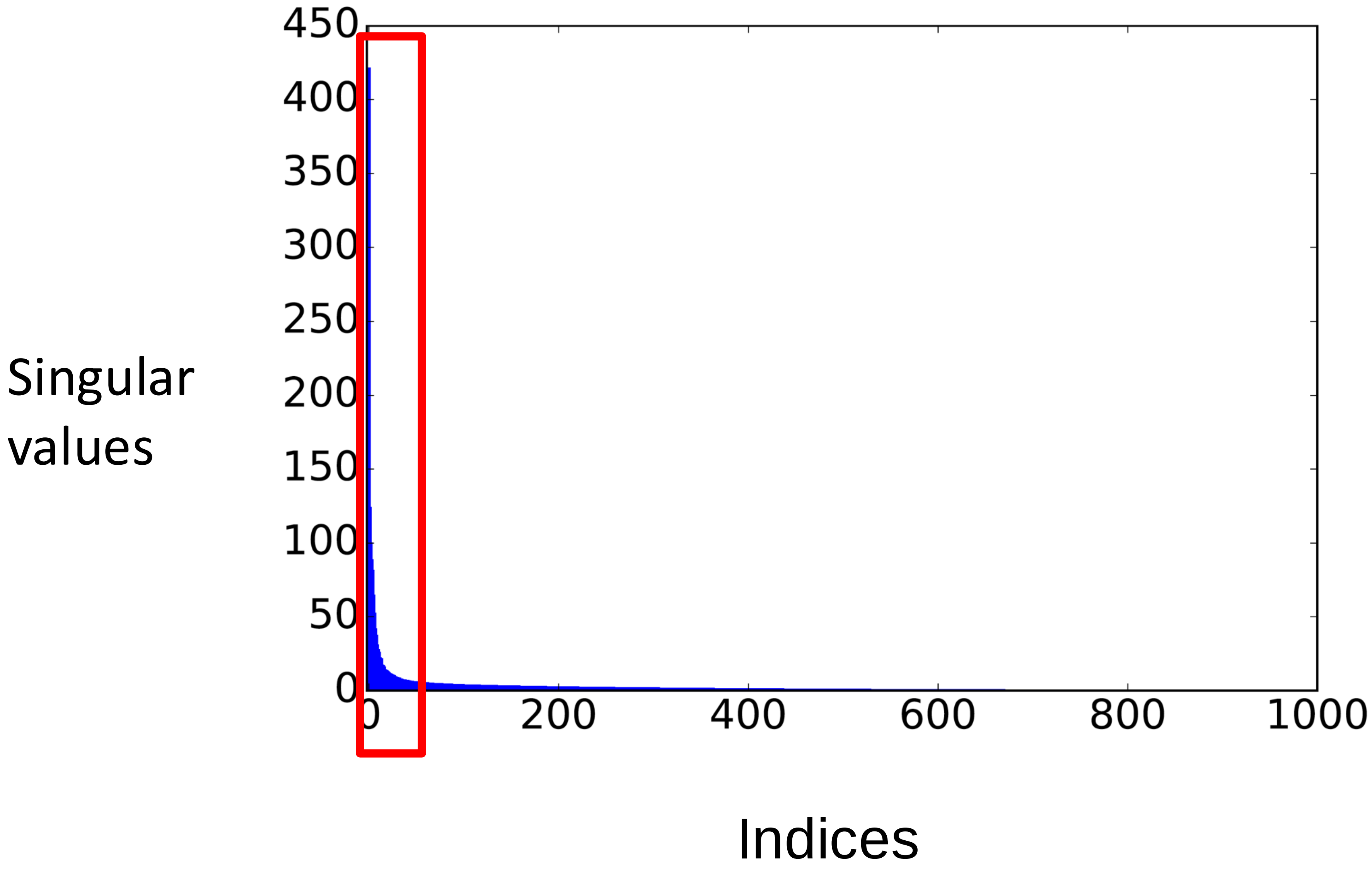
SVD: Example



Singular values

Indices

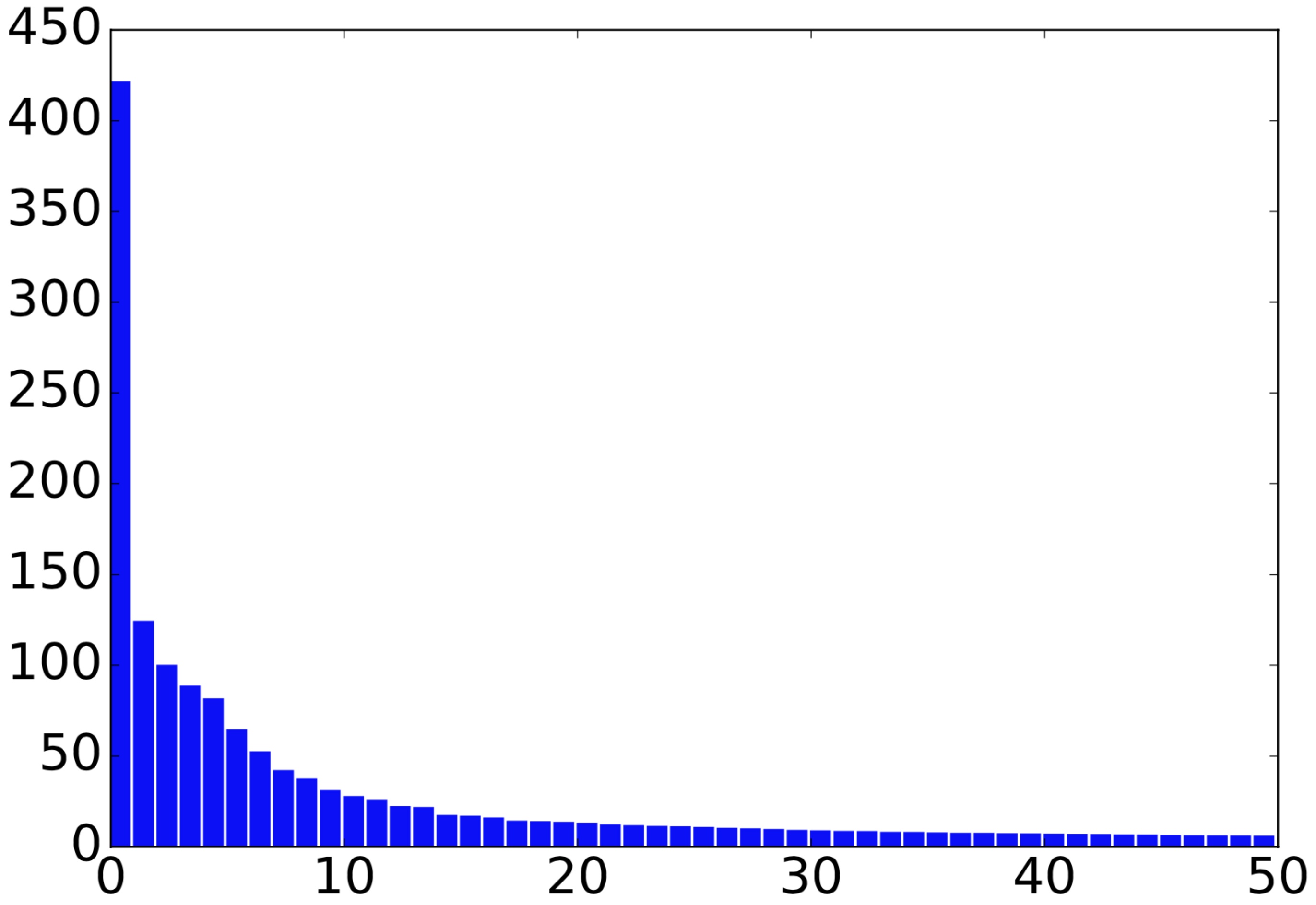
SVD: Example



Singular values
450
400
350
300
250
200
150
100
50
0
0
200
400
600
800
1000
Indices

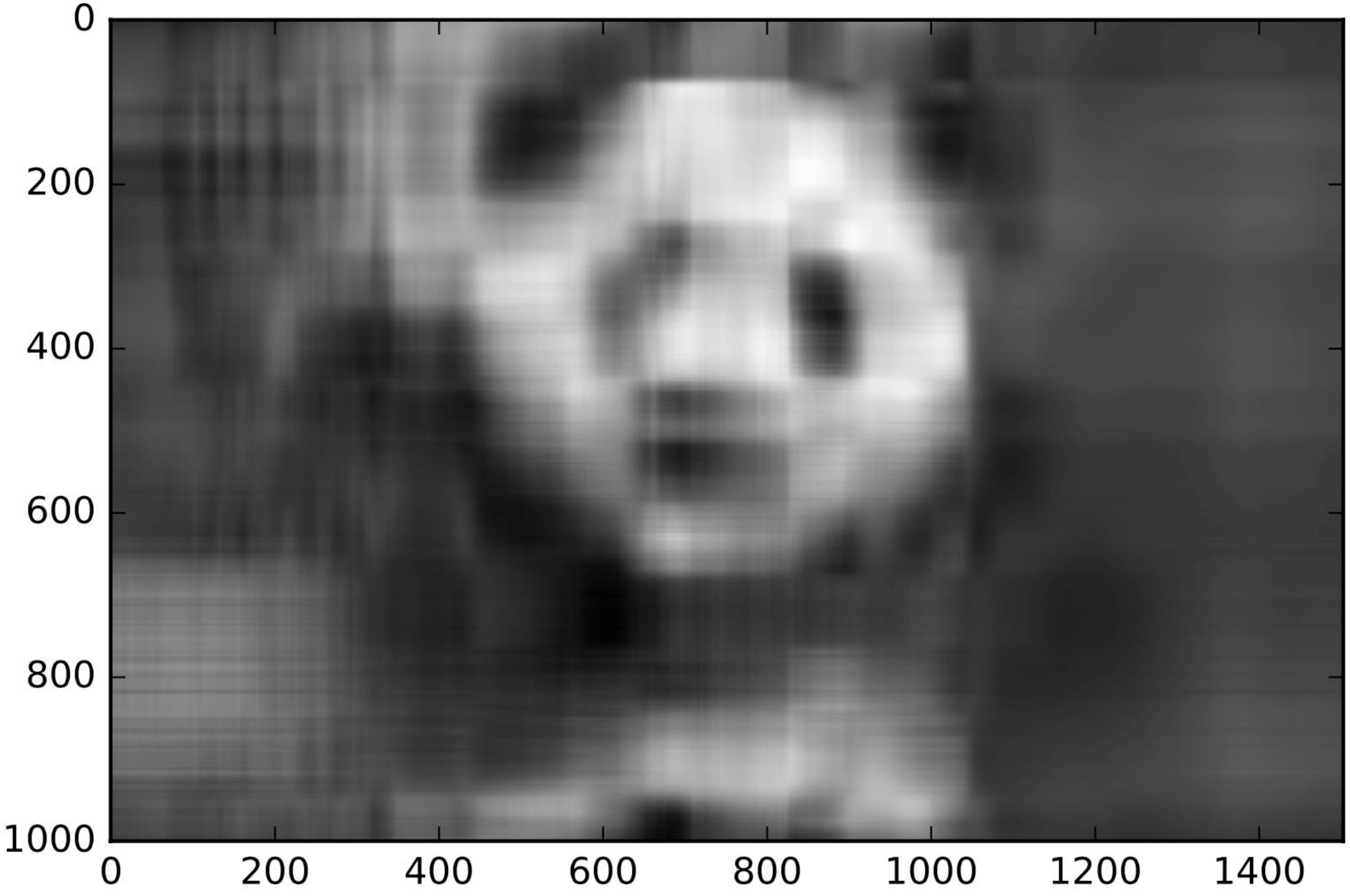
SVD: Example

Top-50
singular
values



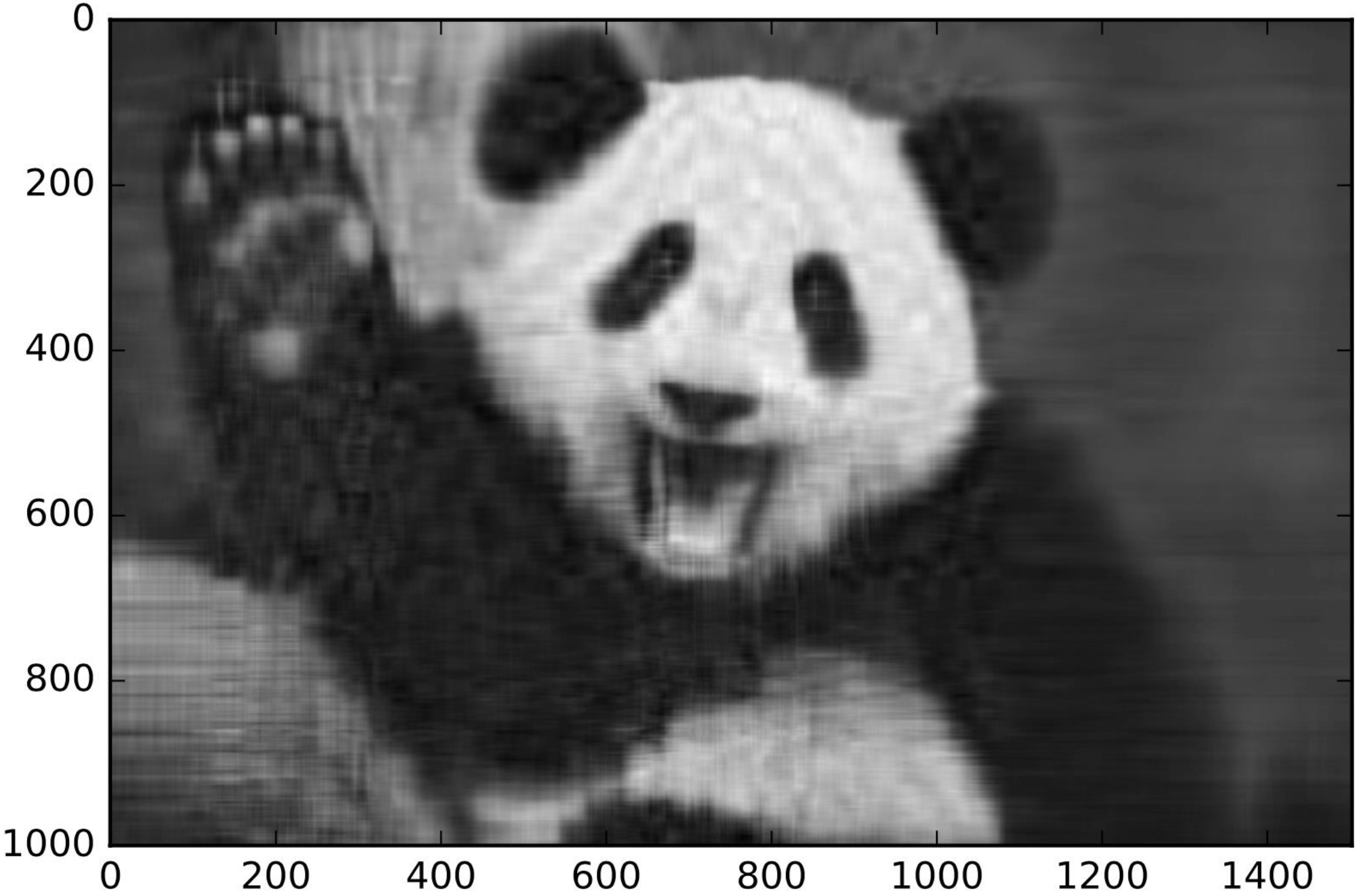
First 50 indices

SVD: Example



Rank-5 truncated SVD

SVD: Example



Rank-20 truncated SVD

SVD: Example

Rank-50 truncated SVD

SVD: Example

Rank-100 truncated SVD

SVD: Example

Original image size (1000 x 1500)

SVD: Example

A high-quality approximation using 10% singular values/vectors!

Rank-100 truncated SVD

SVD: Example

- The original matrix
  - Size: 1000×1500
  - #Entries: 1.5M
- The rank-100 truncated SVD
  - $\mathbf{A}_{100} = \sum_{i=1}^{100} \sigma_i \, \boldsymbol{u}_i \boldsymbol{v}_i^T$
  - Size
    - $\{\sigma_i\}$: 100×1
    - $\{\boldsymbol{u}_i\}$: 100×1000
    - $\{\boldsymbol{v}_i\}$:100×1500
  - #Entries: 0.25M
- Truncated SVD saves **83%** storage

Power Iteration for Computing Truncated SVD

A Property

Theorem. If A = $\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ is the SVD of A, then A^T^A = $\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$

Proof.

- **A^TA =** $\left(\sum_{i=1}^{r} \sigma_i \boldsymbol{v}_i \boldsymbol{u}_i^T \right) \left(\sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T \right)$
- **A^TA =** $\left(\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{u}_i^T \boldsymbol{u}_i \boldsymbol{v}_i^T \right) + \left(\sum_{i \neq j} \sigma_i \sigma_j \boldsymbol{v}_i \boldsymbol{u}_i^T \boldsymbol{u}_j \boldsymbol{v}_j^T \right)$

$$\left(\sum_i \mathbf{x}_i^T \right) \cdot \left(\sum_j \mathbf{x}_j \right) = \sum_i \mathbf{x}_i^T \mathbf{x}_i + \sum_{i \neq j} \mathbf{x}_i^T \mathbf{x}_j$$

A Property

Theorem. If A = $\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ **is the SVD of A, then A^TA** = $\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$

Proof.

- **A^TA =** $\left(\sum_{i=1}^{r} \sigma_i \boldsymbol{v}_i \boldsymbol{u}_i^T\right) \left(\sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T\right)$
- **A^TA =** $\left(\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{u}_i^T \boldsymbol{u}_i \boldsymbol{v}_i^T\right) + \left(\sum_{i \neq j} \sigma_i \sigma_j \boldsymbol{v}_i \boldsymbol{u}_i^T \boldsymbol{u}_j \boldsymbol{v}_j^T\right)$
- **A^TA =** $\left(\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \, 1 \, \boldsymbol{v}_i^T\right) + \left(\sum_{i \neq j} \sigma_i \sigma_j \boldsymbol{v}_i \, 0 \, \boldsymbol{v}_j^T\right)$

Using the properties of orthonormal basis: $\boldsymbol{u}_i^T \boldsymbol{u}_i$=1 and $\boldsymbol{u}_i^T \boldsymbol{u}_j$ =0 for $i \neq j$.

A Property

Theorem. If A = $\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ **is the SVD of A, then A^T^A =** $\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$

Proof.

- **A^TA =** $\left(\sum_{i=1}^{r} \sigma_i \boldsymbol{v}_i \boldsymbol{u}_i^T \right) \left(\sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T \right)$
- **A^TA =** $\left(\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{u}_i^T \boldsymbol{u}_i \boldsymbol{v}_i^T \right) + \left(\sum_{i \neq j} \sigma_i \sigma_j \boldsymbol{v}_i \boldsymbol{u}_i^T \boldsymbol{u}_j \boldsymbol{v}_j^T \right)$
- **A^TA =** $\left(\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \, 1 \, \boldsymbol{v}_i^T \right) + \left(\sum_{i \neq j} \sigma_i \sigma_j \boldsymbol{v}_i \, 0 \, \boldsymbol{v}_j^T \right)$
- **A^TA =** $\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$

A Property

Theorem. If A = $\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ **is the SVD of A, then A^T^A** = $\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$

A^T**A** = $\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$ $\Longrightarrow$ **A**^{T}**A** $\boldsymbol{v}_i = \sigma_i^2 \boldsymbol{v}_i$

Eigenvalue decomposition
of **A**^{T}**A** to obtain (σ_i, v_i)

Efficient Power Iteration Method

Power Iteration for Truncated SVD

Goal: Compute the top 1 eigenvalue/eigenvector of $\mathbf{A}^T\mathbf{A} = \sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$

Algorithm:

1. Randomly initialize a vector $\mathbf{x_0}$ (with unit ℓ_2-norm);
2. Repeat the power iteration: $\mathbf{x_q} \leftarrow \mathbf{A^T A}\ \mathbf{x_{q-1}}$ **and** $\mathbf{x_q} \leftarrow \mathbf{x_q} / \|\mathbf{x_q}\|_2$

2 matrix-vector multiplications

$\mathbf{b} = \mathbf{A}\ \mathbf{x_{q-1}}$

$\mathbf{c} = \mathbf{A^T} b$

Cheap computation

Power Iteration for Truncated SVD

Goal: Compute the top 1 eigenvalue/eigenvector of $\mathbf{A}^T\mathbf{A} = \sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$

Algorithm:

1. Randomly initialize a vector $\mathbf{x}_0$ (with unit ℓ_2-norm);
2. Repeat the power iteration: $\mathbf{x}_q \leftarrow \mathbf{A}^T\mathbf{A}\, \mathbf{x}_{q-1}$ **and** $\mathbf{x}_q \leftarrow \mathbf{x}_q / \|\mathbf{x}_q\|_2$

Convergence analysis ($\mathbf{x}_q$ converges to $\boldsymbol{v}_1$)

- $\boldsymbol{x}_0 = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i$ Every vector can be written as a linear combination of the orthonormal basis
- $\boldsymbol{x}_q \propto (\mathbf{A}^T\mathbf{A})^q\, \boldsymbol{x}_0$ $\quad (\mathbf{A}^T\mathbf{A})^q = \sum_{i=1}^{r} \sigma_i^{2q} \boldsymbol{v}_i \boldsymbol{v}_i^T$
- $\boldsymbol{x}_q \propto \left(\sum_{i=1}^{r} \sigma_i^{2q} \boldsymbol{v}_i \boldsymbol{v}_i^T\right)\left(\sum_{j=1}^{n} \alpha_j \boldsymbol{v}_j\right) = \sum_{i=1}^{r} \alpha_i \sigma_i^{2q} \boldsymbol{v}_i$
- $\boldsymbol{x}_q \propto \sum_{i=1}^{r} \alpha_i \left(\frac{\sigma_i}{\sigma_1}\right)^{2q} \boldsymbol{v}_i = \alpha_1 \boldsymbol{v}_1 + \sum_{i=2}^{r} \alpha_i \left(\frac{\sigma_i}{\sigma_1}\right)^{2q} \boldsymbol{v}_i$

Converge to 0 because $\frac{\sigma_i}{\sigma_1} < 1$

Power Iteration for Truncated SVD

Goal: Compute the top k eigenvalue/eigenvector of $\mathbf{A}^T\mathbf{A} = \sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^T$

Algorithm:

1. Randomly initialize a vector $\mathbf{X_0} \in \mathbb{R}^{n \times k}$
   - Entries are i.i.d. standard Gaussian
2. Orthogonalize the columns: $\mathbf{X_0} \leftarrow \text{orth}(\mathbf{X_0})$;
3. Repeat the power iteration:
   - **i.** $\mathbf{X_q} \leftarrow \mathbf{A^T A}\, \mathbf{X_{q-1}}$;
   - **ii.** $\mathbf{X_q} \leftarrow \text{orth}(\mathbf{X_q})$

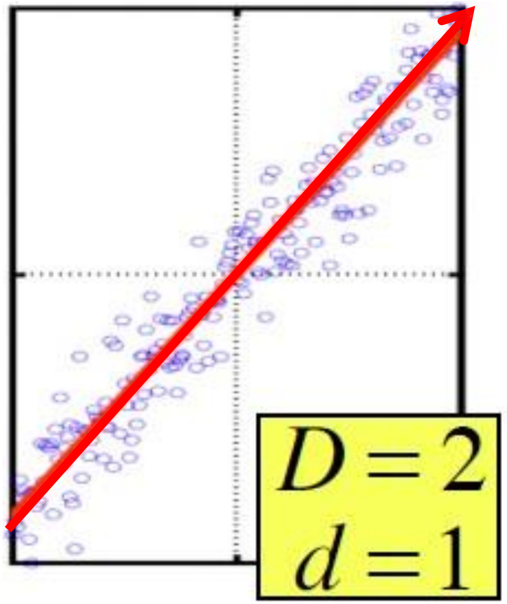
Summary: SVD

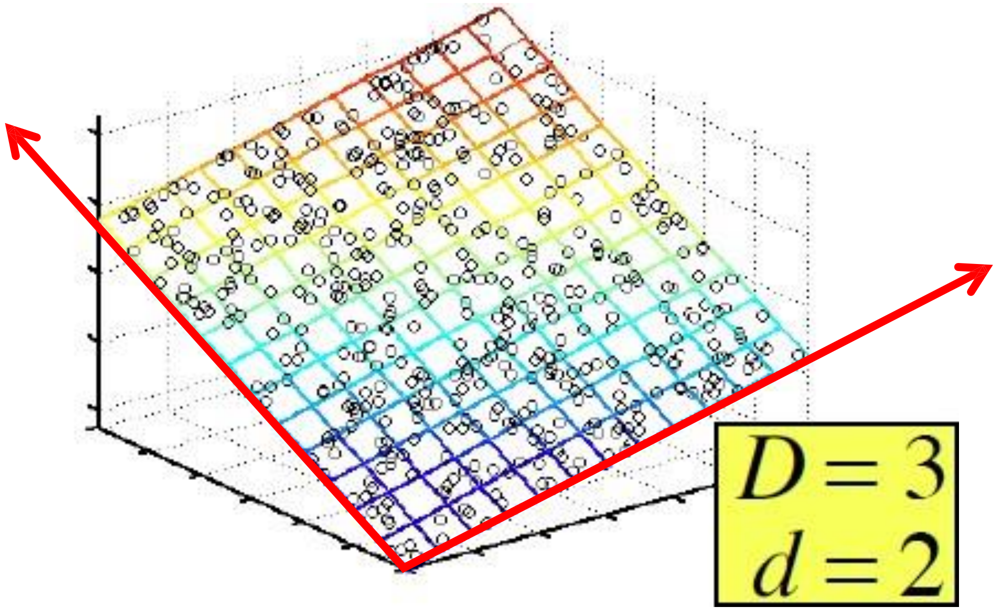
- SVD: Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be anymatrix
  - $\mathbf{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$; r = rank($\mathbf{A}$) ≤ min(m,n).
- Truncated SVD: abandon the bottom singular values/vectors
  - $\mathbf{A}_\mathbf{k} = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$
  - $\mathbf{A}_\mathbf{k}$ is the **best** rank-k approximation to $\boldsymbol{A}$
- Power iteration (algorithm) for computing truncated SVD

Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

Assumption: Data (approximately) lies on a lower dimensional space

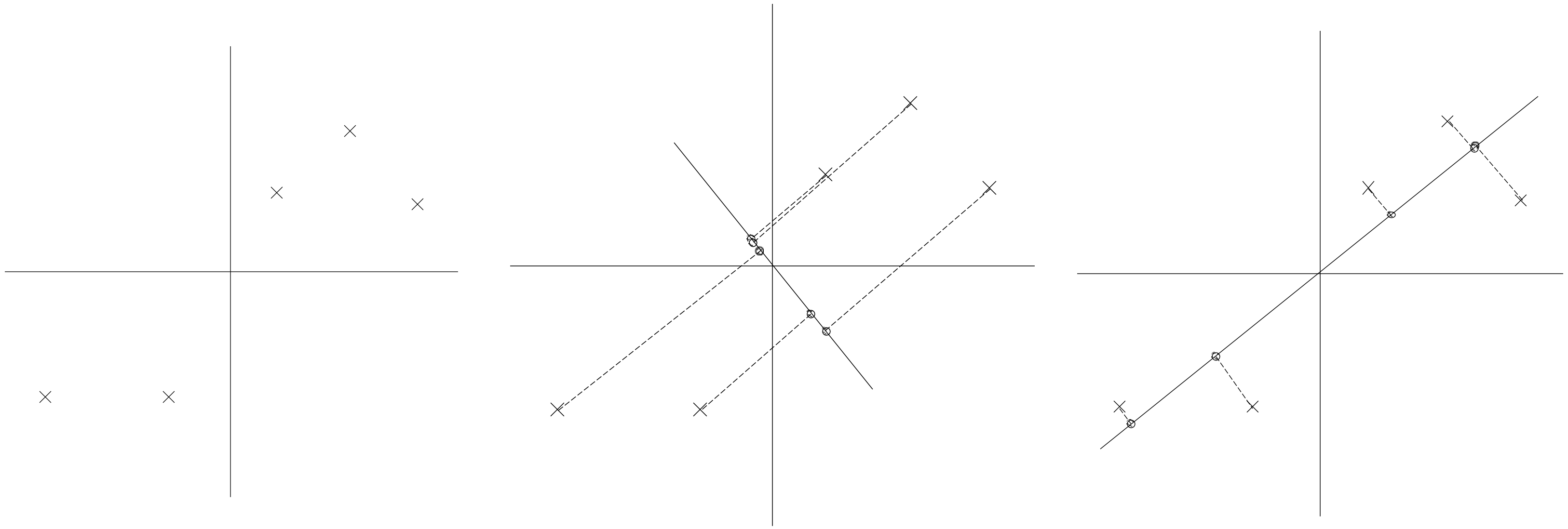




Basis of this subspace are an effective representation of the data

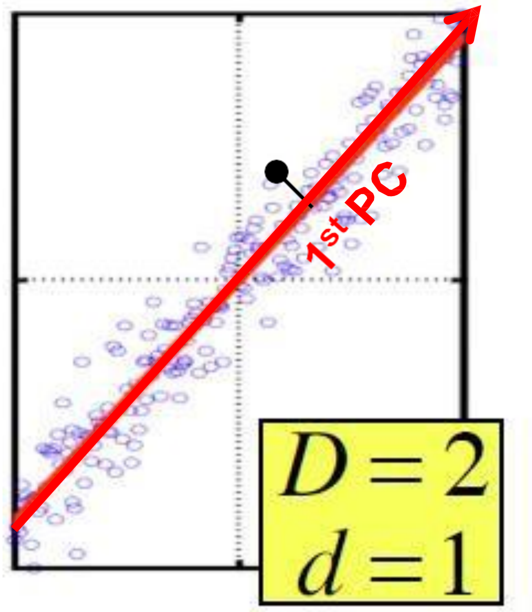
Identifying the basis is known as **Principal Component Analysis**

Which projection is better?



From notes by Andrew Ng

Principal Component Analysis (PCA)

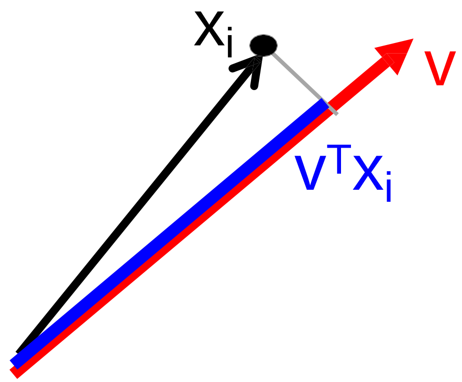


Principal Components (PC) are ***orthogonal*** directions that capture **most of the variance** in the data

1st PC – direction of greatest variability in data

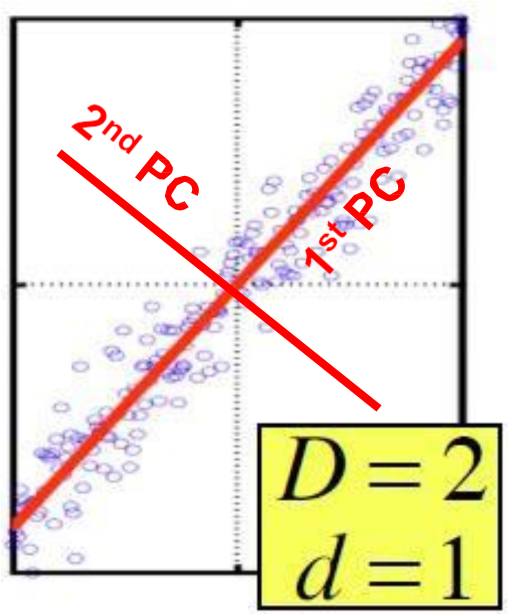
- Projection of data points along 1st PC discriminate the data most along any one direction

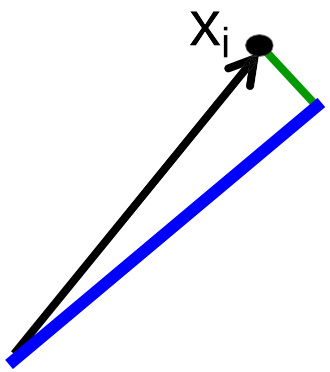
Take a data point x_i (D-dimensional vector)



Projection of x_i onto the 1st PC v is $v^T x_i$

Principal Component Analysis (PCA)





Principal Components (PC) are ***orthogonal*** directions that capture **most of the variance** in the data

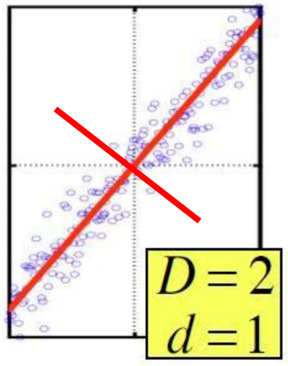
1st PC – direction of greatest variability in data

2nd PC – Next orthogonal direction of greatest variability

- remove all variability in first direction
- then find next direction of greatest variability

And so on …

Principal Component Analysis (PCA)



Data points **X** = [$\boldsymbol{x}_1$, $\boldsymbol{x}_2$,…, $\boldsymbol{x}_n$]: assume data are centered

Let $\boldsymbol{v}_1$,$\boldsymbol{v}_2$, …, $\boldsymbol{v}_d$ denote the principal components

Orthogonal and unit norm $\boldsymbol{v}_i^T\boldsymbol{v}_i$=1 and $\boldsymbol{v}_i^T\boldsymbol{v}_j$ =0 for $i \neq j$

Find vector that **maximizes sample variance** of projection

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}$$

$$\max_{\mathbf{v}} \ \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} \quad \text{s.t.} \quad \mathbf{v}^T\mathbf{v} = 1$$

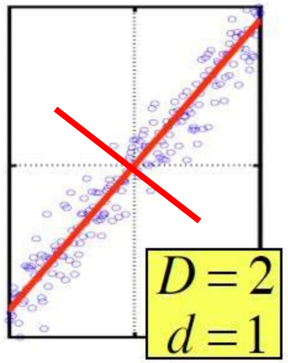
Lagrangian: $\max_{\mathbf{v}} \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} - \lambda\mathbf{v}^T\mathbf{v}$

$$\partial/\partial\mathbf{v} = 0$$

$$(\mathbf{X}\mathbf{X}^T - \lambda\mathbf{I})\mathbf{v} = 0 \qquad \Rightarrow (\mathbf{X}\mathbf{X}^T)\mathbf{v} = \lambda\mathbf{v}$$

Principal Component Analysis (PCA)

$$(\mathbf{X}\mathbf{X}^T)\mathbf{v} = \lambda\mathbf{v}$$



Therefore, v is the eigenvector of sample covariance matrix XX^T

Sample variance of projection = $\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda\mathbf{v}^T\mathbf{v} = \lambda$

Thus, eigenvalue λ denotes the amount of variability captured along that eigenvector

Let eigenvalues $\lambda_1 > \lambda_2 > \lambda_3 > \ldots$

The **1st PC v_1** is the **eigenvector** of the sample covariance matrix XX^T associated with the **largest** eigenvalue λ_1

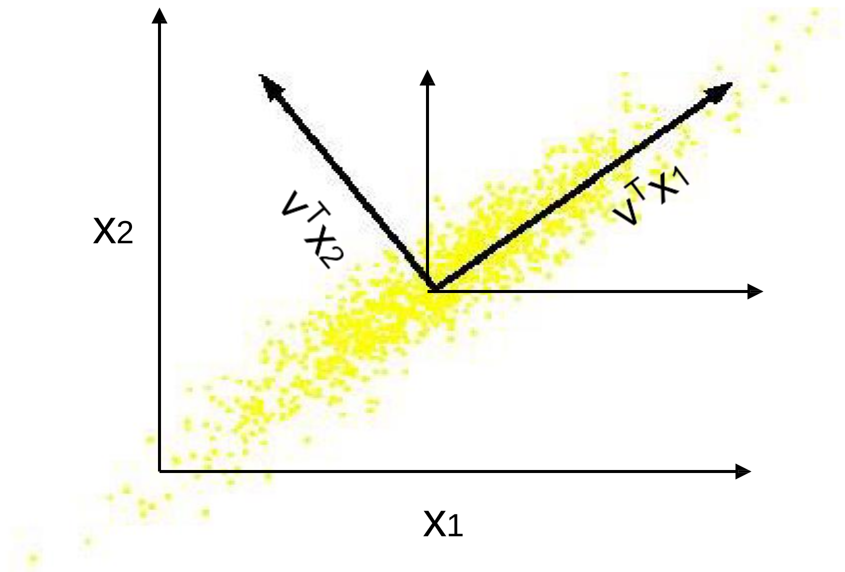
The **2nd PC v_2** is the **eigenvector** of the sample covariance matrix XX^T associated with the **second largest** eigenvalue λ_2

And so on …

Computing the Principal Components

The new basis are the eigenvectors of the sample covariance XX^T of the data

Transformed features are uncorrelated



Geometrical interpretation: centering data by **transition** and then followed by **rotation**

- Linear transformation

Another Interpretation

Maximum Variance Subspace

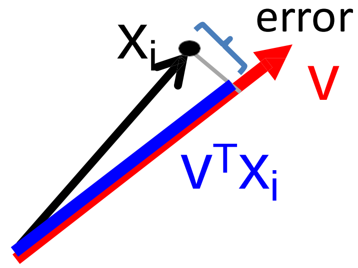
- PCA finds vectors v such that projections on to the vectors capture ***maximum variance in the data***

$$max_{\boldsymbol{v}} \ \frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}$$

Minimum Reconstruction Error

- PCA finds vectors v such that projection on to the vectors yields ***minimum MSE reconstruction***

$$min_{\boldsymbol{v}} \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - (\mathbf{v}^T\mathbf{x}_i)\mathbf{v}\|^2$$

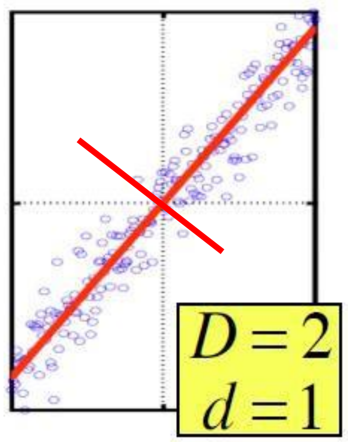


Dimensionality Reduction using PCA

The eigenvalue λ denotes the amount of variability captured along that dimension

Zero eigenvalues indicate no variability along those directions
- Data **exactly** lies on a linear subspace

Keep data projections onto PCs with non-zero eigenvalues
- say v1, …, vd , where d = rank (XX^T)



Original Representation

Data point in the raw D-dimensional space

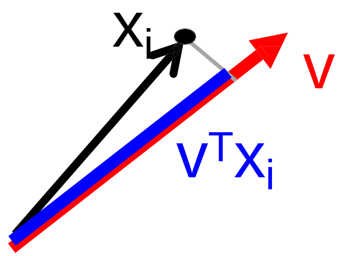
$$\boldsymbol{x_i} = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$$

Transformed representation

Projection matrix $\boldsymbol{V} \in \mathbb{R}^{\boldsymbol{D} \times \boldsymbol{d}}$

(d-dimensional vector)

$$\boldsymbol{V^T x_i} = [\boldsymbol{v_1^T} x_i, \boldsymbol{v_2^T} x_i, \dots, \boldsymbol{v_d^T} x_i]$$

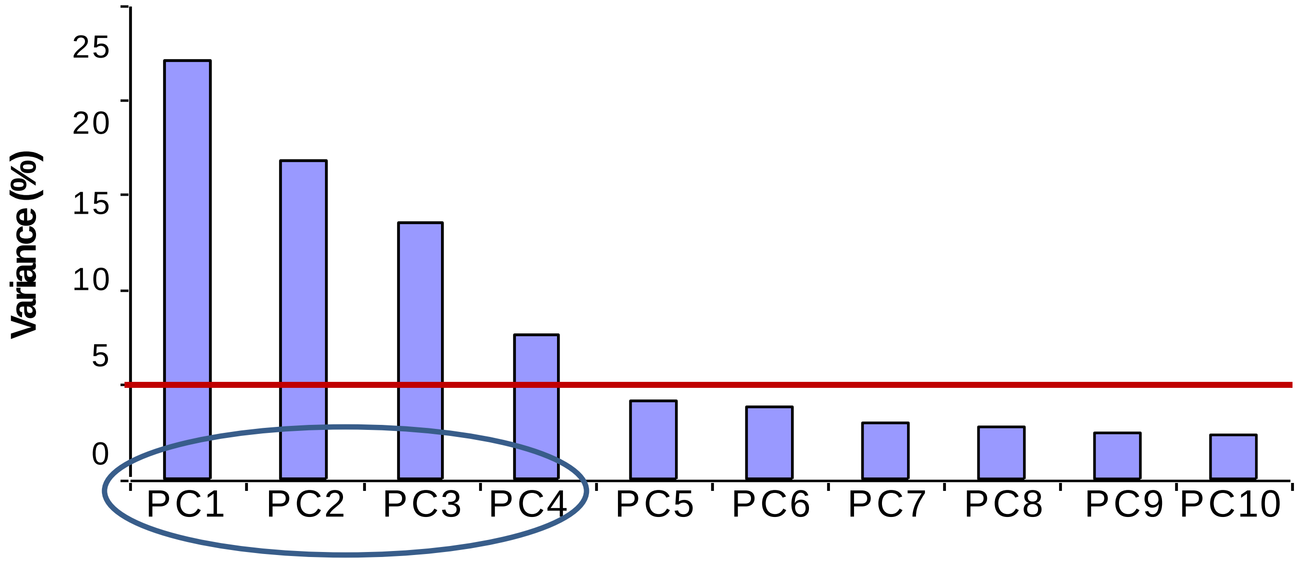


Dimensionality Reduction using PCA

In high-dimensional problem, data usually lies ***near*** a linear subspace, as noise introduces small variability

Only keep data projections onto PCs with **large** eigenvalues

Can *ignore* the components of lesser significance



Might lose some information, but if the eigenvalues are small, don't lose much

Summary: PCA

Project high-dimensional data points into a lower-dimensional space

- **Data points**: $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)$
- **Define mean:** $\underline{\mathbf{x}}$, and let $\underline{X} = X - \underline{\mathbf{x}}$
- **Principal components:** set of orthonormal basis vectors $(\mathbf{v}_1,\ldots,\mathbf{v}_d)$
    - Eigenvalue decomposition: $\underline{\mathbf{X}}\underline{\mathbf{X}}^T\mathbf{v}_i = \lambda_i \mathbf{v}_i$
    - where $<\mathbf{v}_j, \mathbf{v}_j>=1$, and $<\mathbf{v}_i, \mathbf{v}_j>=0$ for $j \neq i$
- **Low-dim representation for $\mathbf{x}_i$:**
    - $\mathbf{z}_i=(z_{i,1},\ldots,z_{i,d})$, where $z_{i,j} = <\mathbf{x}_i - \underline{\mathbf{x}}, \mathbf{v}_j>$

Relationship between SVD and PCA

Centered data points **X** = [$\boldsymbol{x}_1$, $\boldsymbol{x}_2$,…, $\boldsymbol{x}_n$] ∈ $\mathbb{R}^{D\times n}$

$$\text{SVD: } \mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T \qquad \mathbf{U}^T\mathbf{U} = \mathbf{I}_r;\ \mathbf{V}^T\mathbf{V} = \mathbf{I}_r$$

$$\text{PCA: } \mathbf{X}\mathbf{X}^T\mathbf{s}_i = \lambda_i\mathbf{s}_i \Longrightarrow \mathbf{X}\mathbf{X}^T\mathbf{S} = \mathbf{S}\Lambda \Longrightarrow \mathbf{X}\mathbf{X}^T = \mathbf{S}\Lambda\mathbf{S}^T$$

$$\mathbf{s}_i^T\mathbf{s}_i = 1; \mathbf{s}_i^T\mathbf{s}_j = 0, \forall i \neq j \qquad \mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_D] \quad \mathbf{S}^T\mathbf{S} = \mathbf{S}\mathbf{S}^T = \mathbf{I}_D$$

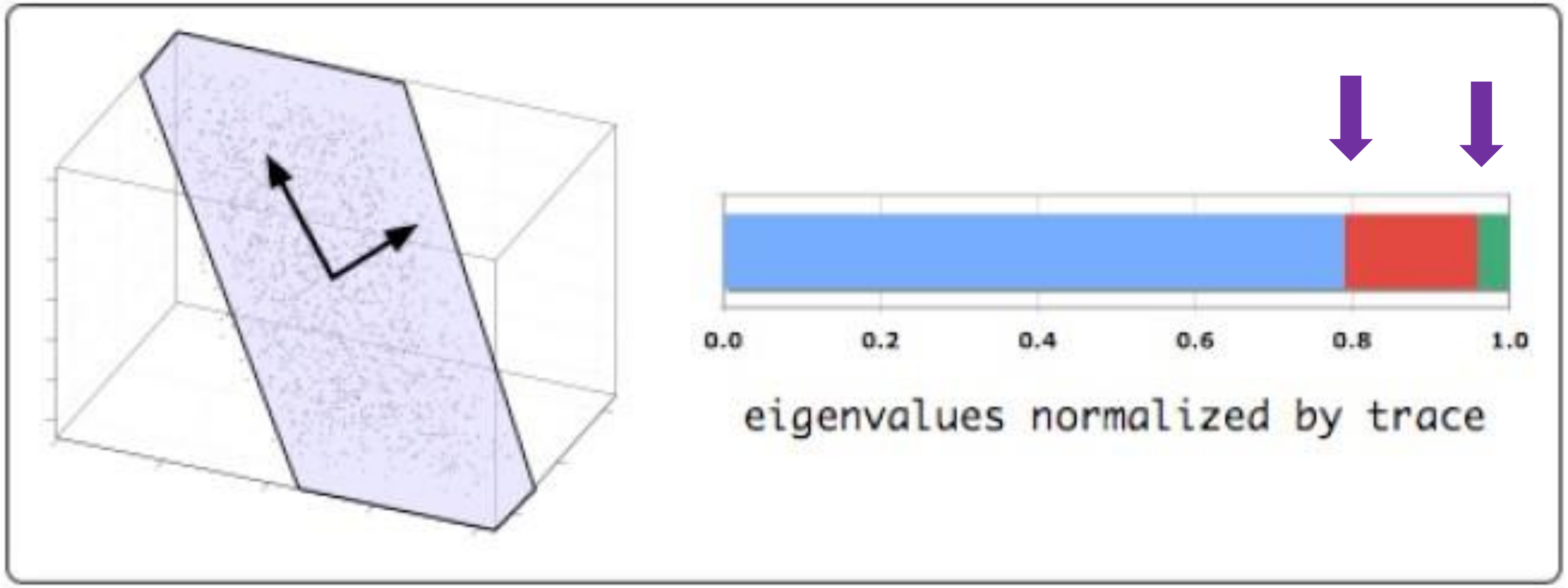
$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma^T\mathbf{U}^T \qquad = \mathbf{U}\Sigma^2\mathbf{U}^T$$

The **left singular vectors (U)** of **X** are the same as the **eigenvectors** (**S**) of $\mathbf{XX}^T$

Similarly, the **eigenvalues** (Λ) of $\mathbf{XX}^T$ are the **squares** of the **singular values (Σ)** of X

Thus, PCA can reduce to computing the SVD of **X** (without forming $\mathbf{XX}^T$)

Example of PCA



Eigenvectors and eigenvalues of covariance matrix for $n=1600$ inputs in $d=3$ dimensions.

PCA for Face Images

The space of all face images

- Each image as vectors of pixel of values

- Image could be high-dimensional
  - E.g., 100 x 100 image = 10,000 dimensions

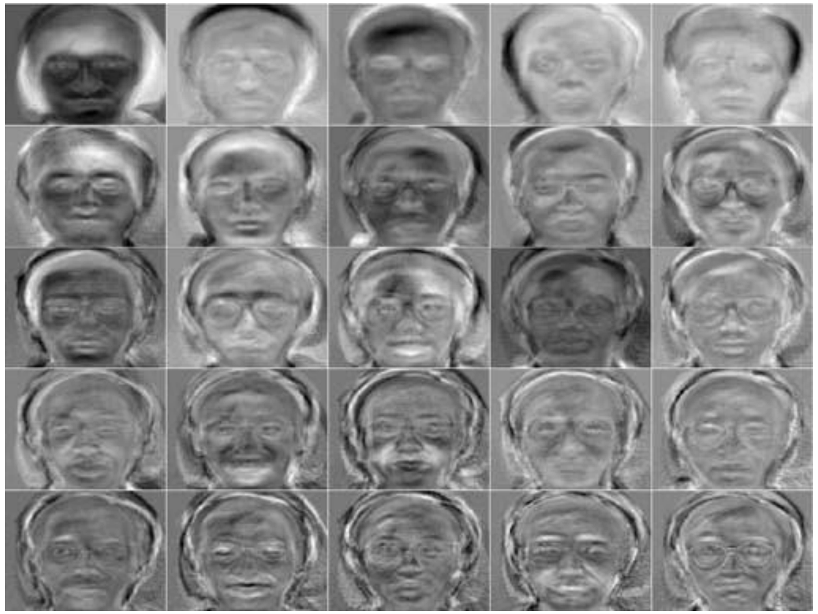
- Few 10,000-dim vectors are valid face images

- We want to effectively model the subspace of face images

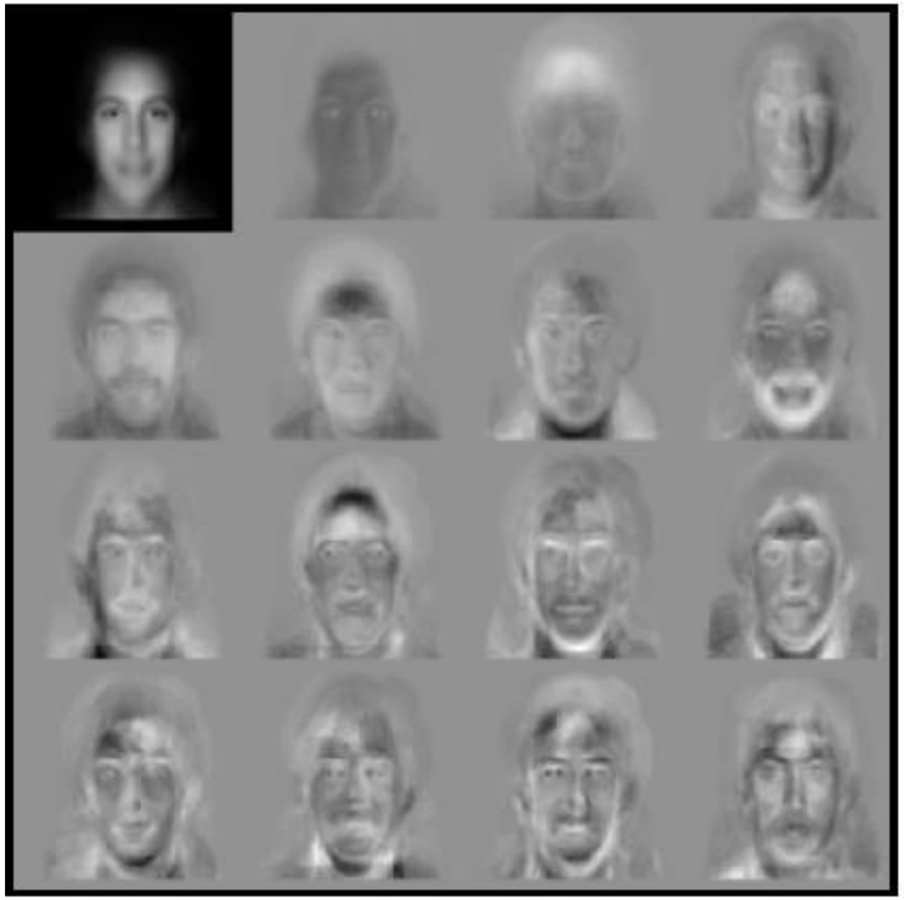
Eigenfaces [Turk, Pentland '91]

Input images

Eigenfaces:
Principal components



Example: faces



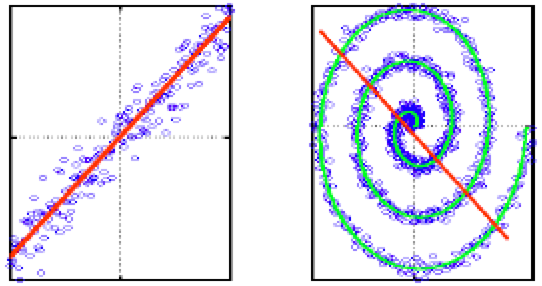
Eigenfaces
from 7562
images:

top left image
is linear
combination
of rest.

Sirovich & Kirby (1987)
Turk & Pentland (1991)

Properties of PCA

- Strengths
  - Eigenvector method
  - No tuning parameters
  - Non-iterative
  - No local optima



- Weaknesses
  - Limited to second order statistics
  - Limited to linear projections

Summary

- Singular Value Decomposition (SVD)
  - **Truncated** SVD as dimensionality reduction
  - **Best** rank-k approximation

- Principal Component Analysis (PCA)
  - **Linearly** project high-dim data points into low-dim space
    - Maximize data variance
    - Minimize data mean square reconstruction

- PCA can be reduced to SVD

Acknowledgement

Some slides are from

Arti Singh (CMU)
https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture20.pdf
Shusen Wang
https://github.com/wangshusen/DeepLearning/blob/master/Slides/5_DR_1.pdf