PRINCIPAL COMPONENT ANALYSIS

CS6140

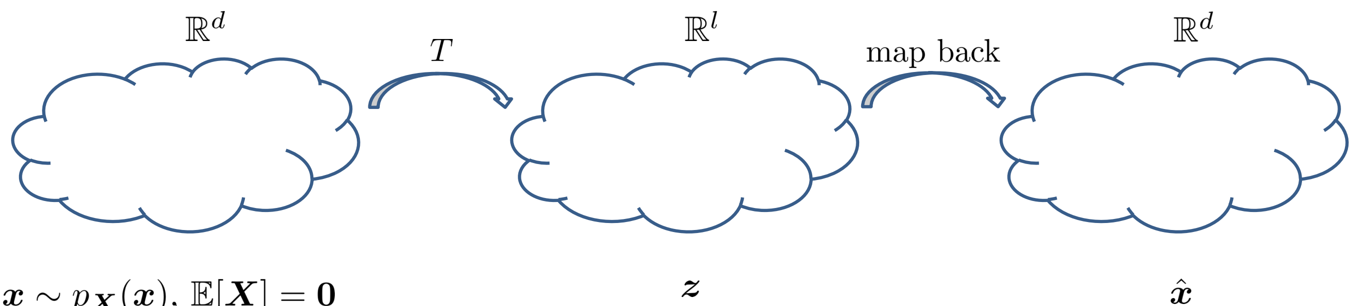
Predrag Radivojac
KHOURY COLLEGE OF COMPUTER SCIENCES
NORTHEASTERN UNIVERSITY

Fall 2024

Problem Formulation

Given: a set of vectors $\{\boldsymbol{x}_i\}_{i=1}^{n}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, sampled from $p_{\boldsymbol{X}}(\boldsymbol{x})$

Objective: find a linear mapping $T : \mathbb{R}^d \to \mathbb{R}^l$, where $l \leq d$, such that the reconstruction of projections back to $\mathbb{R}^d$ is optimal in the mean-squared-error sense.



↑ A minor additional constraint: data is centered.

Linear Mapping

A function $T : \mathbb{R}^d \longrightarrow \mathbb{R}^l$ is a linear mapping if for $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\forall c \in \mathbb{R}$

$$T(\mathbf{x} + \mathbf{y}) = T(\mathbf{x}) + T(\mathbf{y})$$

and

$$T(c\mathbf{x}) = cT(\mathbf{x})$$

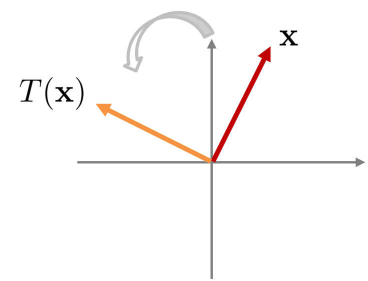
Claim: every linear map T can be represented by an $l \times d$ matrix $\mathbf{T}$ as $T(\mathbf{x}) = \mathbf{T}\mathbf{x}$

Example: rotation by 90° in 2D space.

$$\mathbf{T} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$\mathbf{x} = (2, 4)$$

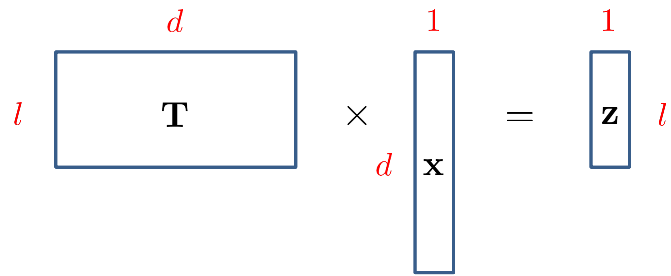
$$T(\mathbf{x}) = \mathbf{T}\mathbf{x} = (-4, 2)$$



Problem Formulation

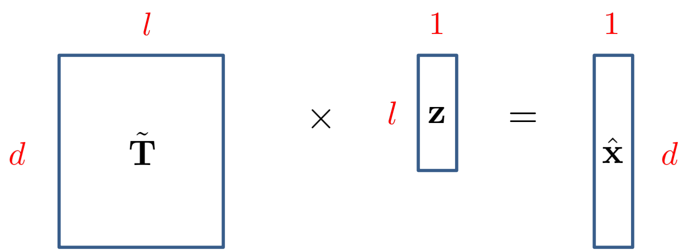
Matrix view: $\mathbf{x} \in \mathbb{R}^{d \times 1}$, $\mathbf{T} \in \mathbb{R}^{l \times d}$. The goal is to find $\mathbf{T}$, $\mathbf{z}$.

$$\mathbf{T}\mathbf{x} = \mathbf{z}$$



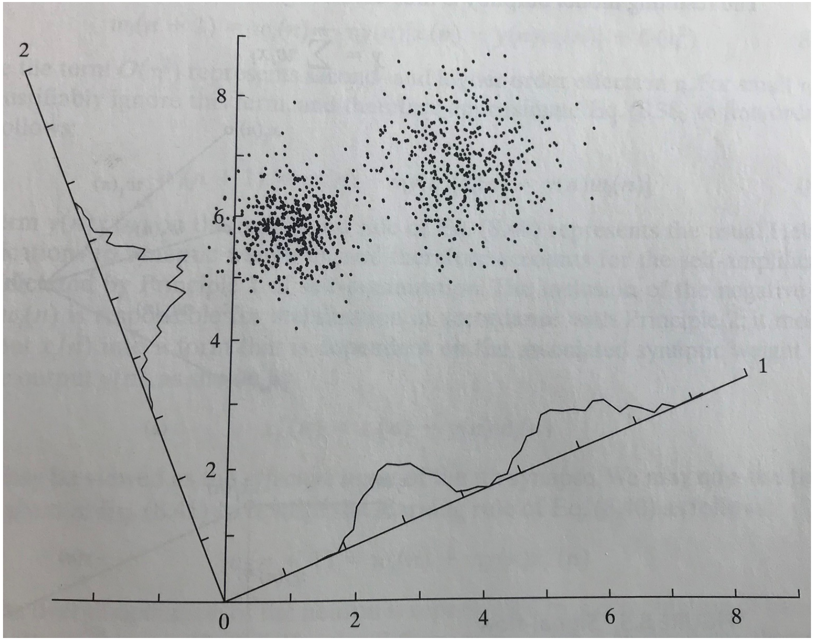
Minimize: $\mathbb{E}[||\boldsymbol{X} - \hat{\boldsymbol{X}}||^2]$

$$\tilde{\mathbf{T}}\mathbf{z} = \hat{\mathbf{x}}$$



It will turn out later that $\tilde{\mathbf{T}}$ is in fact $\mathbf{T}^T$

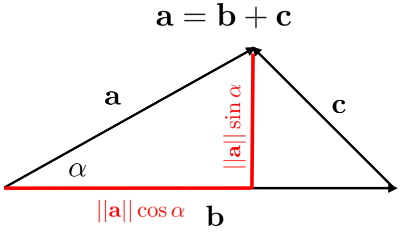
IDEA



Haykin. Neural networks. 1999.

PRELIMINARIES: PROOF FOR COSINE

$$
\begin{aligned}
||\mathbf{c}||^2 &= (||\mathbf{b}|| - ||\mathbf{a}|| \cos\alpha)^2 + (||\mathbf{a}|| \sin\alpha)^2 \\
&= ||\mathbf{b}||^2 - 2||\mathbf{a}|| \cdot ||\mathbf{b}|| \cos\alpha + ||\mathbf{a}||^2 \cos^2\alpha + ||\mathbf{a}||^2 \sin^2\alpha \\
&= ||\mathbf{a}||^2 + ||\mathbf{b}||^2 - 2||\mathbf{a}|| \cdot ||\mathbf{b}|| \cos\alpha
\end{aligned}
$$



$$
\begin{aligned}
||\mathbf{c}||^2 &= \mathbf{c}^T\mathbf{c} \\
&= (\mathbf{a} - \mathbf{b})^T (\mathbf{a} - \mathbf{b}) \\
&= \mathbf{a}^T\mathbf{a} - 2\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{b} \\
&= ||\mathbf{a}||^2 - 2\mathbf{a}^T\mathbf{b} + ||\mathbf{b}||^2
\end{aligned}
$$

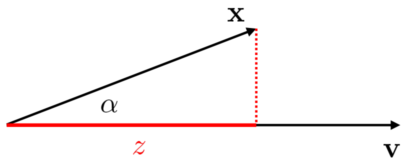
$$\cos(\alpha) = \frac{\mathbf{a}^T\mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||}$$

Combine the two:

$$||\mathbf{a}||^2 - 2\mathbf{a}^T\mathbf{b} + ||\mathbf{b}||^2 = ||\mathbf{a}||^2 + ||\mathbf{b}||^2 - 2||\mathbf{a}|| \cdot ||\mathbf{b}|| \cos\alpha$$

Projection to One Dimension

Let us project a vector $\mathbf{x}$ to a unit vector $\mathbf{v}$. Note: $\mathbf{v}^T\mathbf{v} = 1$ or $||\mathbf{v}|| = 1$.



$$\cos(\alpha) = \frac{z}{||\mathbf{x}||} = \frac{\mathbf{x}^T\mathbf{v}}{||\mathbf{x}|| \cdot ||\mathbf{v}||} \quad \Rightarrow \quad z = \mathbf{x}^T\mathbf{v}$$

Let us project a random vector $\underset{d \times 1}{\boldsymbol{X}} \sim p(\boldsymbol{x})$ to some unit vector $\mathbf{v}$.

$$Z = \boldsymbol{X}^T\mathbf{v} = \mathbf{v}^T\boldsymbol{X}$$

$$\mathbb{E}[Z] = \mathbf{v}^T\mathbb{E}[\boldsymbol{X}] = 0$$

$$\mathbb{E}[Z^2] = \mathbb{E}[\mathbf{v}^T\boldsymbol{X}\boldsymbol{X}^T\mathbf{v}] = \mathbf{v}^T\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^T]\mathbf{v} = \mathbf{v}^T\underset{d \times d}{\boldsymbol{\Sigma}}\mathbf{v} \quad \Rightarrow \quad \mathbb{V}[Z] = \mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v}$$

Projection to One Dimension

For a set of vectors, let us find a unit vector $\mathbf{v}$ so that the projection has maximum variance $\mathbb{V}[Z] = \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v}$.

Objective: Given $\boldsymbol{\Sigma}$, find $\mathbf{v}$ to maximize variance of the projection.

$$\max \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v} \qquad \text{s.t.} \quad \mathbf{v}^T \mathbf{v} = 1$$

$$L(\mathbf{v}, \lambda) = \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v} + \lambda(1 - \mathbf{v}^T \mathbf{v}) \qquad \overset{\text{Solve}}{\Rightarrow} \qquad \boldsymbol{\Sigma} \mathbf{v} = \lambda \mathbf{v}$$

The eigenvalue problem

Projection to *d* Dimensions

Consider now projecting to d orthogonal vectors:

$$\mathbf{\Sigma V} = \mathbf{V \Lambda}$$ $\leftarrow$ matrix version

where $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_d]$, with $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ $\leftarrow$ because $\mathbf{V}$ is orthogonal

$$\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_d\}, \text{ with } \lambda_1 \geq \lambda_2 \dots \geq \lambda_d$$

Let us re-write: $\mathbf{V}^T\mathbf{\Sigma V} = \mathbf{\Lambda}$

$$\mathbf{v}_i^T \mathbf{\Sigma} \mathbf{v}_j = \begin{cases} \lambda_i & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$ $\leftarrow$ variance of projection Z_i

Transformation

Let us express the i-th projection as $z_i = \mathbf{v}_i^T \mathbf{x} = \mathbf{x}^T \mathbf{v}_i$

Thus,

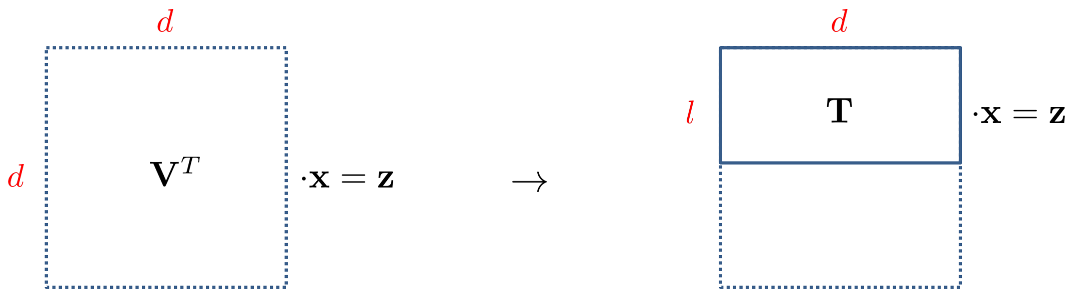
$$\mathbf{z} = (z_1, z_2, \ldots, z_d) = (\mathbf{v}_1^T \mathbf{x}, \mathbf{v}_2^T \mathbf{x}, \ldots, \mathbf{v}_d^T \mathbf{x}) = \mathbf{V}^T \mathbf{x} = \sum_{i=1}^{d} x_i \mathbf{v}_i^T$$

Let us reconstruct $\mathbf{x}$ now. Remember, $\mathbf{V}^{-1} = \mathbf{V}^T$.

$$\mathbf{x} = \mathbf{V}\mathbf{z} = \sum_{i=1}^{d} z_i \mathbf{v}_i$$

Dimensionality Reduction

Let us now keep the first l components of $\mathbf{z}$.



Matrix view:

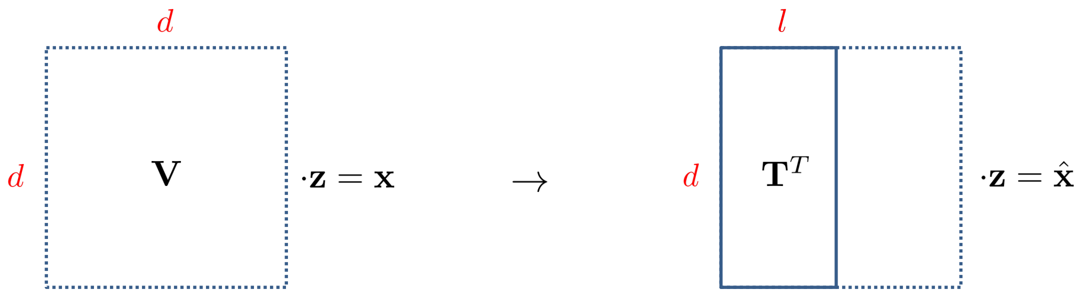
$\ddot{\mathbf{V}}$ is $\mathbf{V}$ reduced to l columns

$\ddot{\mathbf{V}}_{d\times l}$ reminds us of $\ddot{\mathbf{V}}$'s dimensions

$$\underset{n\times d}{\mathbf{Z}} = \mathbf{X}\mathbf{V} \quad \rightarrow \quad \underset{n\times l}{\mathbf{Z}} = \mathbf{X}\ddot{\mathbf{V}}_{d\times l} = \mathbf{X}\mathbf{T}^T$$

Reconstruction

Let us reconstruct $\mathbf{x}$ now:



Matrix view:

$$\overset{\substack{n \times d \\ \downarrow}}{\mathbf{X}} = \mathbf{Z}\mathbf{V}^T \quad \rightarrow \quad \begin{aligned} \hat{\mathbf{X}} &= \mathbf{Z}\ddot{\mathbf{V}}^T_{d \times l} \\ &= \mathbf{Z}\mathbf{T} \end{aligned}$$

Reconstruction Error

Let us reconstruct $\mathbf{x}$ now:

$$\hat{\mathbf{x}} = \sum_{i=1}^{l} z_i \mathbf{v}_i$$

The error vector $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$ is now

$$\mathbf{e} = \sum_{i=l+1}^{d} z_i \mathbf{v}_i$$

because

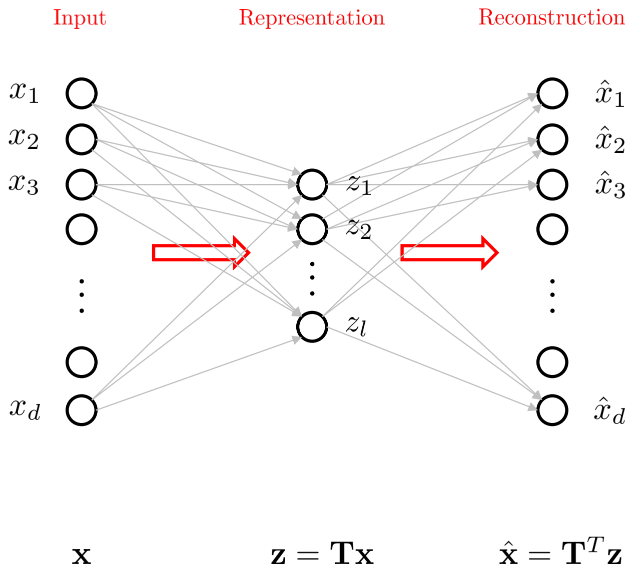
$$\mathbf{x} - \hat{\mathbf{x}} = \sum_{i=1}^{d} z_i \mathbf{v}_i - \sum_{i=1}^{l} z_i \mathbf{v}_i$$

We now have

$$\mathbb{E}[\boldsymbol{X} - \hat{\boldsymbol{X}}] = \mathbf{0} - \textstyle\sum_{i=1}^{l} \mathbb{E}[Z_i]\mathbf{v}_i = \mathbf{0}$$

$$\mathbb{E}[||\boldsymbol{X} - \hat{\boldsymbol{X}}||^2] = \textstyle\sum_{i=l+1}^{d} \mathbf{v}_i^T \boldsymbol{\Sigma} \mathbf{v}_i = \sum_{i=l+1}^{d} \lambda_i$$ ← proved later

PRINCIPAL COMPONENT ANALYSIS AS REPRESENTATION LEARNING



Relationship with Singular Value Decomposition (SVD)

$$\overset{n \times d}{\downarrow}$$

Every matrix $\mathbf{X}$ has a SVD: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$.

$\mathbf{U}$ = orthogonal, $n \times n$
$\mathbf{S}$ = diagonal, $n \times d$
$\mathbf{V}^T$ = orthogonal, $d \times d$

In MATLAB: $[\mathrm{U}, \mathrm{S}, \mathrm{V}] = \mathrm{svd}(\mathrm{X})$

Let's look at $\mathbf{X}^T\mathbf{X}$

$$\mathbf{X}^T\mathbf{X} = (\mathbf{U}\mathbf{S}\mathbf{V}^T)^T(\mathbf{U}\mathbf{S}\mathbf{V}^T) = \mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T.$$

Recall, $\frac{1}{n-1}\mathbf{X}^T\mathbf{X}$ is the estimated covariance matrix when $\mathbf{X}$ is normalized

$$\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} = \frac{1}{n-1}\mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T.$$

$$\mathbf{\Lambda} = \frac{1}{n-1}\mathbf{S}^T\mathbf{S}.$$ $\leftarrow$ eigenvalue matrix

EIGENDECOMPOSITION VS. SINGULAR VALUE DECOMPOSITION

Eigendecomposition: $\frac{1}{n-1}\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$

Singular value decomposition: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$

In MATLAB: $[\mathrm{V}, \Lambda] = \mathrm{eig}(\Sigma)$

$[\mathrm{U}, \mathrm{S}, \mathrm{V}] = \mathrm{svd}(\mathrm{X})$

Q: Is matrix $\mathbf{V}$ exactly the same in both?

A: Should be but not necessarily. Vectors in $\mathbf{V}$ can have opposite directions.

Depends on the software we use.

Computational Complexity

We were solving the following system:

$$\mathbf{\Sigma V} = \mathbf{V \Lambda}$$

where $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_d]$, with $\mathbf{V}^T\mathbf{V} = \mathbf{I}$

$$\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_d\}, \text{ with } \lambda_1 \geq \lambda_2 \dots \geq \lambda_d$$

Total complexity: $O(d^3 + nd^2)$

- computing the covariance matrix ($\mathbf{\Sigma}$): $O(nd^2)$
- computing eigenvectors ($\mathbf{V}$) and eigenvalues ($\mathbf{\Lambda}$): $O(d^3)$

Singular value decomposition takes $O(\min\{nd^2, dn^2\})$

Strang. Introduction to linear algebra. Wellesley-Cambridge Press, 2021.

Handling High-Dimensional Data

Consider a centered data matrix $\overset{n \times d}{\overset{\downarrow}{\mathbf{X}}}$, where $d \gg n$.

$\overset{d \times d}{\overset{\downarrow}{\mathbf{\Sigma}}}$ cannot fit in memory!

Pick now any eigenvalue λ and the corresponding eigenvector $\mathbf{v}$

$$\begin{aligned} \hat{\mathbf{\Sigma}}\mathbf{v} &= \lambda\mathbf{v} \\ \tfrac{1}{n-1}\mathbf{X}^T\mathbf{X}\mathbf{v} &= \lambda\mathbf{v} \\ \tfrac{1}{n-1}\mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{v} &= \lambda\mathbf{X}\mathbf{v} \end{aligned} \quad \Longrightarrow \quad \tfrac{1}{n-1}\overset{n \times n}{\overset{\downarrow}{\mathbf{X}\mathbf{X}^T}} \underbrace{\mathbf{X}\mathbf{v}}_{\substack{\uparrow \\ n \times 1}} = \lambda \underbrace{\mathbf{X}\mathbf{v}}_{\substack{\uparrow \\ n \times 1}}$$

Note: $\mathbf{X}$ is still column-normalized

Handling High-Dimensional Data

$$\frac{1}{n-1}\overset{\overset{n \times d}{\downarrow}}{\mathbf{X}}\mathbf{X}^T \underbrace{\mathbf{X}\ddot{\mathbf{V}}}_{\mathbf{Z}} = \underbrace{\mathbf{X}\ddot{\mathbf{V}}}_{\mathbf{Z}}\mathbf{\Lambda}$$

Note: $d \gg n$, so we reduce $\mathbf{V}$ to $\ddot{\mathbf{V}}_{d \times l}$.

Eigenvalues of $\frac{1}{n-1}\mathbf{X}^T\mathbf{X}$ are the same as eigenvalues of $\frac{1}{n-1}\mathbf{X}\mathbf{X}^T$

There are at most n nonzero eigenvalues, for both $\frac{1}{n-1}\mathbf{X}^T\mathbf{X}$ and $\frac{1}{n-1}\mathbf{X}\mathbf{X}^T$

Solution:

$$\frac{1}{n-1}\mathbf{X}\mathbf{X}^T\overset{\overset{n \times n}{\downarrow}}{\mathbf{W}} = \mathbf{W}\mathbf{\Lambda}$$

Note: we can reduce $\mathbf{W}$ to $\ddot{\mathbf{W}}_{n \times l}$.

$$\frac{1}{n-1}\mathbf{X}^T\mathbf{X}\underbrace{\mathbf{X}^T\mathbf{W}}_{\mathbf{V}'} = \underbrace{\mathbf{X}^T\mathbf{W}}_{\mathbf{V}'}\mathbf{\Lambda}$$

The norm of each column of $\mathbf{W}$ is 1, but not for $\mathbf{X}^T\mathbf{W}$. $\leftarrow$ we centered $\mathbf{X}$ not $\mathbf{X}^T$

$\mathbf{V} \leftarrow \text{normalize}(\mathbf{V}')$ so that column norms are 1.

Handling High-Dimensional Data

Normalizing $\mathbf{V}'$ has a closed-form formula: $\mathbf{V} = \underset{n\times d}{\mathbf{X}^T}\,\underset{n\times n}{\mathbf{W}}\cdot\text{diag}\left\{\sqrt{\underset{n\times n}{\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}}}\right\}$

$$\frac{1}{n-1}\mathbf{X}\mathbf{X}^T\underbrace{\mathbf{X}\ddot{\mathbf{V}}}_{\mathbf{Z}} = \underbrace{\mathbf{X}\ddot{\mathbf{V}}}_{\mathbf{Z}}\boldsymbol{\Lambda}$$

Algorithm:

Solve $\frac{1}{n-1}\mathbf{X}\mathbf{X}^T\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}$ to find $\boldsymbol{\Lambda}$ and $\mathbf{W}$

Keep l columns of $\mathbf{W}$ to obtain $\ddot{\mathbf{W}}_{n\times l}$

$\ddot{\mathbf{V}}_{d\times l} = \mathbf{X}^T\ddot{\mathbf{W}}_{n\times l}\cdot\text{diag}\left\{\sqrt{\ddot{\mathbf{W}}_{n\times l}^T\mathbf{X}\mathbf{X}^T\ddot{\mathbf{W}}_{n\times l}}\right\}$

$\mathbf{Z} = \mathbf{X}\ddot{\mathbf{V}}_{d\times l}$

Additional considerations:

What if $\mathbf{X}$ is sparse with huge d and we cannot center it?

What if some columns of $\mathbf{X}$ are constant?

Application: Eigenfaces

Given: a set of n images $\mathbf{X}_{n \times d}$, where each row is a flattened matrix.

Find: transformation matrix $\mathbf{T}_{l \times d}$.

X

$\longleftarrow$ a sample from Yale Faces B set, with $n = 5000+$ images of 28 subjects

$\leftarrow$ each row is a sample of 5 images for the same subject

$\leftarrow$ each image is processed to a 48×42 matrix, so $d = 48 \cdot 42 = 2016$

https://www.face-rec.org

Mean image:

First 20 eigenvectors, shown as scaled matrices:

T

Sirovich & Kirby. Low-dimensional procedure for the characterization of human faces. *J Opt Soc Am A*, 1987. Turk & Pentland. Eigenfaces for recognition. *J Cogn Neurosci*, 1991.

Reconstruction Error

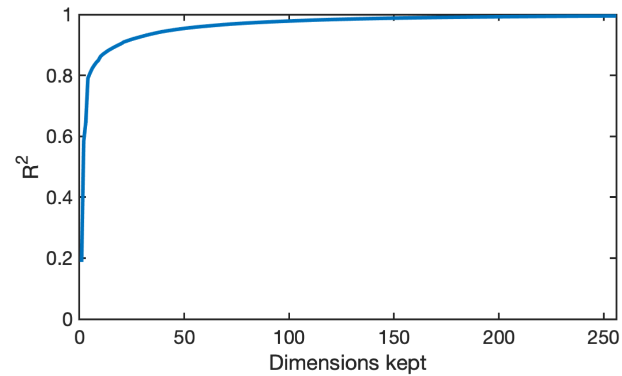
Yale Faces B data set

$$R^2 = 1 - \frac{\sum_{i=1}^{n} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2}{\sum_{i=1}^{n} ||\mathbf{x}_i - \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i||^2}$$

Original Reconstructed

$l =$ 1 2 4 8 16 32 64 128 256

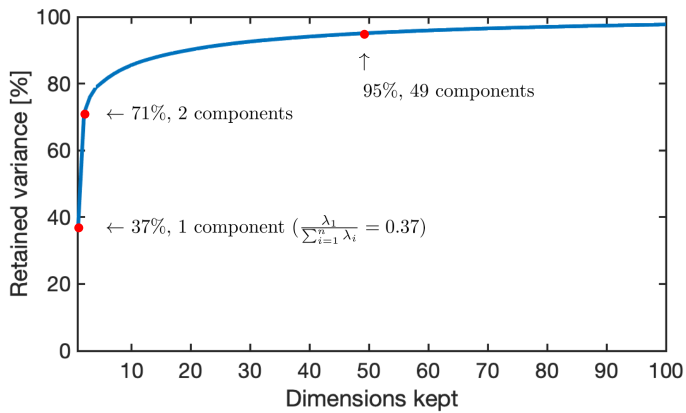
↑ $R^2 = 0.1878$ ↑ $R^2 = 0.8906$ ↑ $R^2 = 0.9949$



Note: reconstruction error is measured on the "training" set

How Many Components to Keep?

Yale Faces B data set



99%, 181 components

99.9%, 556 components

It is often better to specify the percent of ratained variance, and not l.

APPLICATION: LATENT SEMANTIC ANALYSIS FOR DOCUMENT RETRIEVAL

Given: an $n \times d$ text document matrix $\mathbf{X}$

n = number of documents
d = dictionary size

Find: latent semantic spaces for document retrieval and term similarity.

Semantic space = space where "terms and documents that are closely associated are placed near one another" (Deerwester et al., 1990).

	access	*document*	*retrieval*	*information*	*theory*	*database*	*indexing*	*computer*	REL	MATCH
Doc 1	x	x	x			x	x		R	
Doc 2				x*	x			x*		M
Doc 3			x	x*				x*	R	M

R = relevant
M = matched

Query: "IDF in *computer*-based *information* look-up"

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \approx \ddot{\mathbf{U}}_{n\times l}\ddot{\mathbf{S}}_{l\times l}\ddot{\mathbf{V}}^T_{d\times l}$$

$$\mathbf{X}^T = \mathbf{V}\mathbf{S}^T\mathbf{U}^T \approx \ddot{\mathbf{V}}_{d\times l}\ddot{\mathbf{S}}^T_{l\times l}\ddot{\mathbf{U}}^T_{n\times l}$$

Term similarities: $\mathbf{X}^T\mathbf{X} \approx \ddot{\mathbf{V}}\ddot{\mathbf{S}}^T\ddot{\mathbf{S}}\ddot{\mathbf{V}}^T$

Document similarities: $\mathbf{X}\mathbf{X}^T \approx \ddot{\mathbf{U}}\ddot{\mathbf{S}}\ddot{\mathbf{S}}^T\ddot{\mathbf{U}}^T$

Deerwester et al. Indexing by latent semantic analysis. *J Am Soc Inf Sci*, 1990.

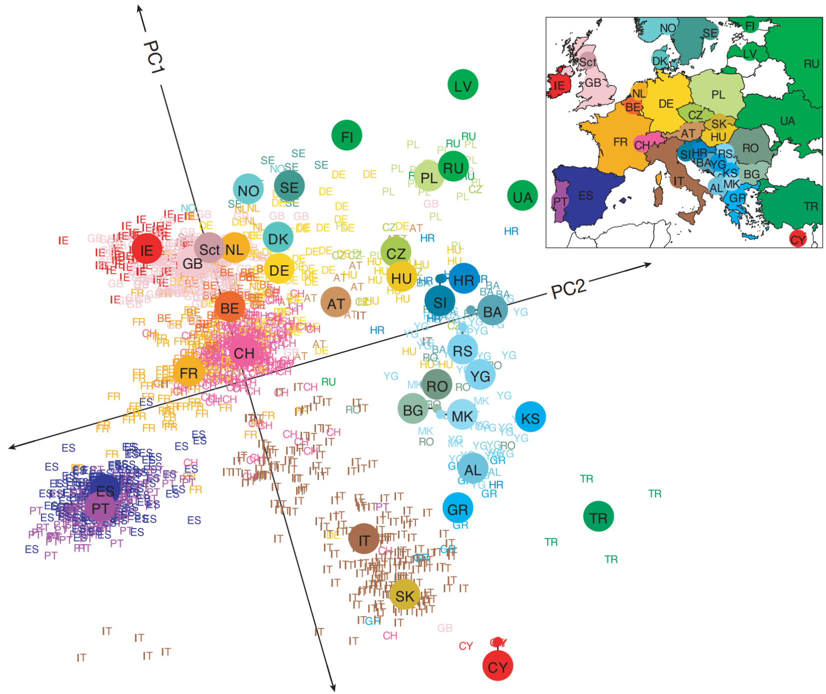
Application: Genomic Data Visualization

$$\mathbf{X}_{n \times d} = \text{sparse matrix}$$

$n = 3192$ subjects

$d = 500568$ genomic loci

$l = 2$ principal directions



Novembre et al. Genes mirror geography within Europe. *Nature*, 2008.

Kernel PCA

Consider high-dimensional data. We had $\overset{\overset{n \times n}{\downarrow}}{\boldsymbol{\Gamma}} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T$, where $\Gamma_{ij} = \frac{1}{n-1}\mathbf{x}_i^T\mathbf{x}_j$.

A mapping $\boldsymbol{\varphi} : \mathcal{X} \to \mathcal{F}$ allows us to use any domain for inputs; i.e., $x \in \mathcal{X}$.

$$K_{ij} = \boldsymbol{\varphi}(x_i)^T\boldsymbol{\varphi}(x_j)$$

$\leftarrow$ **K** is positive semi-definite

Problem: $\frac{1}{n}\sum \mathbf{x}_i = \mathbf{0}$, but $\frac{1}{n}\sum \boldsymbol{\varphi}(x_i)$ is arbitrary.

$$\mathbf{K} \leftarrow \mathbf{K} - \mathbf{1}_n\mathbf{K} - \mathbf{K}\mathbf{1}_n + \mathbf{1}_n\mathbf{K}\mathbf{1}_n$$

$\leftarrow$ proved later

$\mathbf{1}_n$ = an $n \times n$ matrix where each element is $\frac{1}{n}$

Schölkopf et al. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*, 1998.

Handling Test Data with Kernel PCA

Given: training set $\{x_i\}_{i=1}^{n}$ and test set $\{t_i\}_{i=1}^{m}$;

i.e., an $n \times n$ training kernel matrix $\mathbf{K}$ and $m \times n$ test matrix $\mathbf{K}^{\text{test}}$.

$$K_{ij} = \boldsymbol{\varphi}(x_i)^T \boldsymbol{\varphi}(x_j)$$

$$K_{ij}^{\text{test}} = \boldsymbol{\varphi}(t_i)^T \boldsymbol{\varphi}(x_j)$$

Centering:

$$\underset{m \times n}{\underset{\downarrow}{\mathbf{K}^{\text{test}}}} \leftarrow \mathbf{K}^{\text{test}} - \mathbf{1}'_n \underset{n \times n}{\underset{\downarrow}{\mathbf{K}}} - \mathbf{K}^{\text{test}} \mathbf{1}_n + \mathbf{1}'_n \mathbf{K} \mathbf{1}_n$$

$\mathbf{1}'_n$ = an $m \times n$ matrix where each element is $\frac{1}{n}$

$\mathbf{1}_n$ = an $n \times n$ matrix where each element is $\frac{1}{n}$

Schölkopf et al. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*, 1998.

Differences between Kernel PCA and PCA

Kernel PCA vs. PCA

- KPCA offers many choices of similarity functions
  - $\mathbf{K}$ must be symmetric positive semi-definite
- input space for KPCA need not be $\mathbb{R}^d$
  - KPCA can directly operate on seqeunces, strings, graphs
- classification accuracy often improved over PCA, given l
- KPCA allows $l > d$, PCA does not
- loss of interpretability with KPCA
  - cannot easily visualize eigenvectors for images
  - requires separate optimization
- computing time a problem for KPCA when n is large
- additional numerical problems with KPCA
  - centering may cause that $K_{ij} \neq K_{ji}$ which gives complex $\mathbf{\Lambda}$

Schölkopf et al. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*, 1998.

APPENDIX: PROOF #1

Proof for the squared norm of the error vector:

$$\begin{aligned}\mathbb{E}[||(\boldsymbol{X}-\hat{\boldsymbol{X}})||^2] &= \mathbb{E}[(\boldsymbol{X}-\hat{\boldsymbol{X}})^T(\boldsymbol{X}-\hat{\boldsymbol{X}})] \\ &= \mathbb{E}[\boldsymbol{X}^T\boldsymbol{X}] - 2\mathbb{E}[\boldsymbol{X}^T\hat{\boldsymbol{X}}] + \mathbb{E}[\hat{\boldsymbol{X}}^T\hat{\boldsymbol{X}}]\end{aligned}$$

We investigate one of these terms

$$\mathbb{E}[\hat{\boldsymbol{X}}^T\hat{\boldsymbol{X}}] = \mathbb{E}[\sum_{i=1}^{l} Z_i \mathbf{v}_i^T \cdot \sum_{j=1}^{l} Z_j \mathbf{v}_j] = \mathbb{E}[\sum_{i=1}^{l} Z_i^2 \mathbf{v}_i^T \mathbf{v}_i] = \mathbb{E}[\sum_{i=1}^{l} Z_i^2]$$

because $\mathbf{v}_i^T\mathbf{v}_j = 0$ when $i \neq j$ and $\mathbf{v}_i^T\mathbf{v}_j = 1$ when $i = j$. This makes a double sum above a single sum.

Appendix: Proof #1

We similarly have

$$\mathbb{E}[\boldsymbol{X}^T\boldsymbol{X}] = \mathbb{E}[\sum_{i=1}^{d} Z_i \mathbf{v}_i^T \cdot \sum_{j=1}^{d} Z_j \mathbf{v}_j] = \mathbb{E}[\sum_{i=1}^{d} Z_i^2]$$

$$\mathbb{E}[\boldsymbol{X}^T\hat{\boldsymbol{X}}] = \mathbb{E}[\sum_{i=1}^{d} Z_i \mathbf{v}_i^T \cdot \sum_{j=1}^{l} Z_j \mathbf{v}_j] = \mathbb{E}[\sum_{i=1}^{l} Z_i^2]$$

Finally, we have

$$\mathbb{E}[(\boldsymbol{X} - \hat{\boldsymbol{X}})^T(\boldsymbol{X} - \hat{\boldsymbol{X}})] = \sum_{i=1}^{d} \lambda_i - 2\sum_{i=1}^{l} \lambda_i + \sum_{i=1}^{l} \lambda_i = \sum_{i=l+1}^{d} \lambda_i$$

Q.E.D.

Appendix: Proof #2

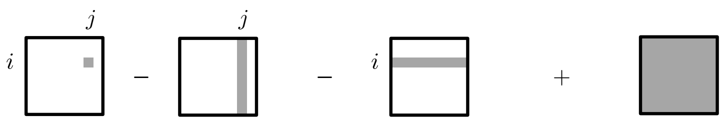
Proof for the kernel normalized in the feature space:

$$K_{ij} \leftarrow \left(\boldsymbol{\varphi}(x_i) - \frac{1}{n}\sum_{k=1}^{n}\boldsymbol{\varphi}(x_k)\right)^T\left(\boldsymbol{\varphi}(x_j) - \frac{1}{n}\sum_{l=1}^{n}\boldsymbol{\varphi}(x_l)\right)$$

$\leftarrow$ centering in feature space

$$= \boldsymbol{\varphi}(x_i)^T\boldsymbol{\varphi}(x_j) - \frac{1}{n}\sum_{l=1}^{n}\boldsymbol{\varphi}(x_i)^T\boldsymbol{\varphi}(x_l) - \frac{1}{n}\sum_{l=1}^{n}\boldsymbol{\varphi}(x_k)^T\boldsymbol{\varphi}(x_j) + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}\boldsymbol{\varphi}(x_k)^T\boldsymbol{\varphi}(x_l)$$

$$= K_{ij} - \frac{1}{n}\sum_{k=1}^{n}K_{kj} - \frac{1}{n}\sum_{l=1}^{n}K_{il} + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}K_{kl}$$



Matrix form:

$$\mathbf{K} \leftarrow \mathbf{K} - \mathbf{1}_n\mathbf{K} - \mathbf{K}\mathbf{1}_n + \mathbf{1}_n\mathbf{K}\mathbf{1}_n$$

$\mathbf{1}_n$ = an $n \times n$ matrix where each element is $\frac{1}{n}$

Q.E.D.