

K SCHOOL

K SCHOOL

K SCHOOL

K SCHOOL

INTRODUCCIÓN AL ÁLGEBRA LINEAL

www.kschool.com

Deep Learning

An MIT Press book in preparation

Ian Goodfellow, Aaron Courville, and Yoshua Bengio

- [Table of Contents](#)
- [Acknowledgements](#)
- [Notation](#)
- [1 Introduction](#)

◦ [Part I: Applied Math and Machine Learning Basics](#)

- [2 Linear Algebra](#)
- [3 Probability and Information Theory](#)
- [4 Numerical Computation](#)
- [5 Machine Learning Basics](#)

[Part II: Modern Practical Deep Networks](#)

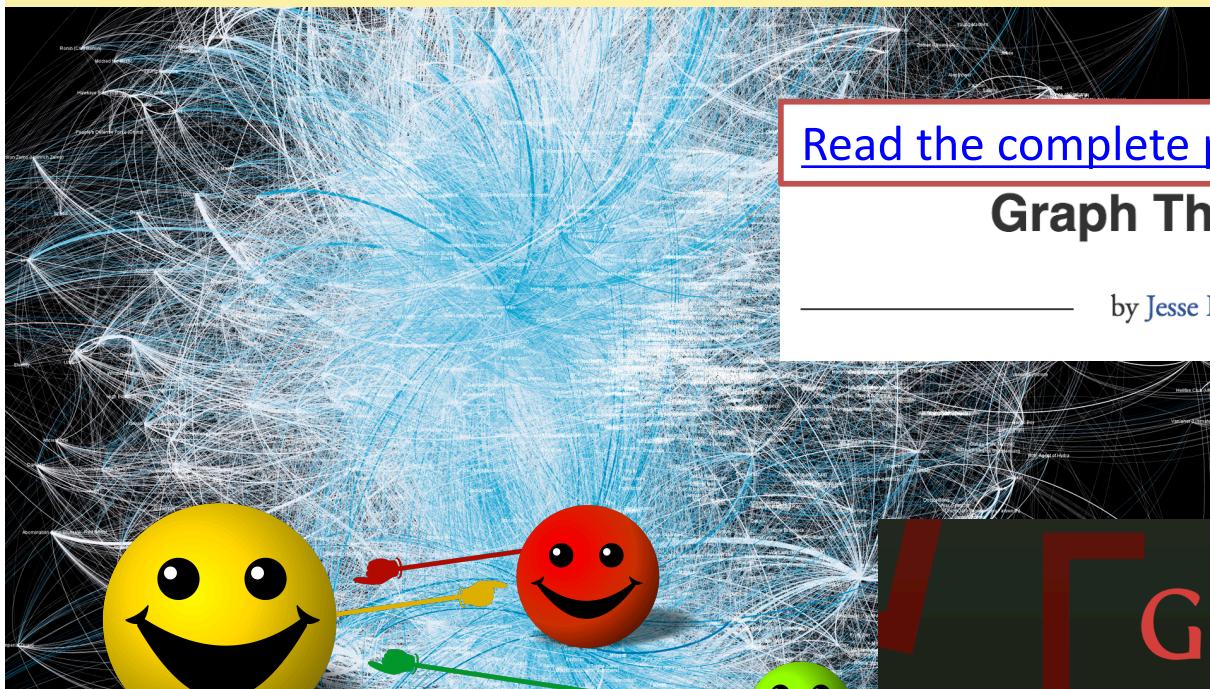
- [6 Feedforward Deep Networks](#)
- [7 Regularization](#)
- [8 Optimization for Training Deep Models](#)
- [9 Convolutional Networks](#)
- [10 Sequence Modeling: Recurrent and Recursive Nets](#)
- [11 Practical Methodology](#)
- [12 Applications](#)

[Part III: Deep Learning Research](#)

- [13 Linear Factor Models and Auto-Encoders](#)
- [14 Representation Learning](#)
- [15 The Manifold Perspective on Representation Learning](#)
- [16 Structured Probabilistic Models for Deep Learning](#)
- [17 Monte Carlo Methods](#)
- [18 Confronting the Partition Function](#)
- [19 Approximate Inference](#)
- [20 Deep Generative Models](#)

- [Bibliography](#)
- [Index](#)

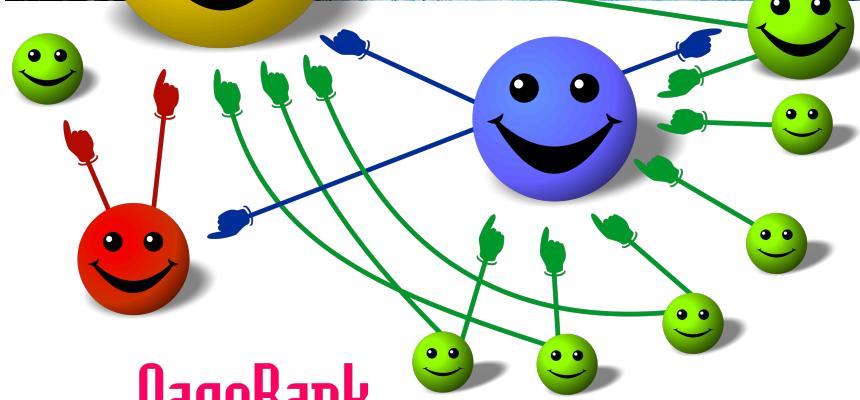
- IPython Notebook: 2005-2011
- pandas: 2008-2009
- scikit-learn: 2007
- NumPy: 2006
- matplotlib: 2002
- IPython: 2001
- Numarray: 2001
- SciPy: 1999
- Numeric: 1995
- High performance linear algebra, image processing, optimization via NumPy, optimized C++, FORTRAN
- Large structured data via HDF5, memmap
- Out of core processing, streaming & realtime
- Distributed computing via MPI, IPython Parallel, etc.
- GPU & heterogenous via OpenCL, PyCUDA, others
- Massive adoption in research, national labs, industry (engineering, finance, etc.)



[Read the complete post](#)

Graph Theory: Part III (Facebook)

by Jesse Farmer on Wednesday, August 24, 2011



PageRank

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu



Prezi

[See the talk!](#)

Graph Theory: In my Internet?

Facebooking & Googling
with Eigenvectors

Jen Mathews

Básicos: vectores y matrices

Máster de Data Science

www.kschool.com





Trasponer vectores y matrices:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}; \quad \mathbf{x}^T = [x_1, x_2, \dots, x_n]$$

$$\mathbf{A} = (A_{ij}) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}; \quad \mathbf{A}^T = \mathbf{B} = (B_{ij}) = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}$$



Algunas propiedades básicas:

- Productos y normas:

$$\mathbf{x}^T \cdot \mathbf{y} = \sum_i x_i \cdot y_i$$

$$\mathbf{x}^T \cdot \mathbf{x} = \sum_i x_i^2 = \|\mathbf{x}\|_2^2; \quad \mathbf{x}^T \cdot \mathbf{y} = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos \theta$$

$$\mathbf{C} = \mathbf{AB} \Rightarrow c_{i,j} = (\mathbf{A} \cdot \mathbf{B})_{i,j} = \sum_k a_{i,k} \cdot b_{k,j}$$

$$\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p \right)^{1/p}; \quad \|\mathbf{A}\|_F = \sqrt{\sum_{i,j} a_{i,j}^2}$$

- Productos de matrices:

$$\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$$

$$\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$$

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$$

- Matriz unitaria, inversa y sistema de ecuaciones:

$$\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{I}_n \mathbf{x} = \mathbf{x}$$

$$\mathbf{A} \in \mathbb{R}^{n \times n}, |\mathbf{A}| \neq 0 \Rightarrow \mathbf{A}^{-1} \mathbf{A} = \mathbf{A} \mathbf{A}^{-1} = \mathbf{I}_n$$

Sean $\mathbf{x} \in \mathbb{R}^m, \mathbf{b} \in \mathbb{R}^n, \mathbf{A} \in \mathbb{R}^{n \times m}$:

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{A}^{-1} \mathbf{Ax} = \mathbf{A}^{-1} \mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{b}$$



Más sobre matrices (cuadradas!)

- Traza:

$$Tr(\mathbf{A}) = \sum_i a_{i,i}$$

$$Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$$

$$Tr(\mathbf{A}^{-1}) = 1 / Tr(\mathbf{A})$$

$$Tr(\mathbf{XY}) = \sum_i \left(\sum_k x_{i,k} \cdot y_{k,i} \right) = \sum_k \left(\sum_i x_{i,k} \cdot y_{k,i} \right) = Tr(\mathbf{YX})$$

$$Tr(\mathbf{A}^T \mathbf{B}) = Tr(\mathbf{B}^T \mathbf{A}) = Tr(\mathbf{AB}^T) = Tr(\mathbf{BA}^T)$$

$$Tr(\mathbf{ABC}) = Tr(\mathbf{BCA}) = Tr(\mathbf{CBA})$$

$$Tr(\mathbf{P}^{-1} \mathbf{AP}) = Tr(\mathbf{APP}^{-1}) = Tr(\mathbf{A})$$

- Determinante:

$$\det(\mathbf{A}) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^n a_{i,\sigma_i}$$

← [wikipedia!](#)

$$\det(\mathbf{A}) = \det(\mathbf{A}^T)$$

$$\det(\mathbf{A}^{-1}) = \det(\mathbf{A})^{-1}$$

$$\det(\mathbf{AB}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$$



Algunas matrices especiales:

- Matriz diagonal:

$$\mathbf{D} / d_{i,j} = 0 \quad \forall i \neq j$$

$$\mathbf{D}^{-1} = \text{diag}(\mathbf{v})^{-1} = \text{diag}\left(\left[1/v_1, \dots, 1/v_n\right]^T\right)$$

$$Tr(\mathbf{D}) = \sum v_i; \quad \det(\mathbf{D}) = \prod v_i$$

$$\mathbf{D} = \text{diag}(\mathbf{v}) = \begin{bmatrix} v_1 & 0 & \cdots & 0 \\ 0 & v_2 & 0 & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & v_n \end{bmatrix}$$

- Matriz simétrica: $\mathbf{S} = \mathbf{S}^T \Rightarrow s_{i,j} = s_{j,i}$

- Matriz ortogonal:

$$\mathbf{M}^T \mathbf{M} = \mathbf{M} \mathbf{M}^T = \mathbf{I} \Rightarrow \mathbf{M}^{-1} = \mathbf{M}^T$$

$$\det(\mathbf{M}) = 1$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1^T \\ \vdots \\ \mathbf{m}_n^T \end{bmatrix}; \mathbf{M}^T = [\mathbf{m}_1, \dots, \mathbf{m}_n]; \mathbf{m}_i^T \mathbf{m}_j = \delta_{i,j}$$

- Matriz semi-definida positiva: $\forall \mathbf{x} \in R^n, \mathbf{x}^T \mathbf{P} \mathbf{x} \geq 0$



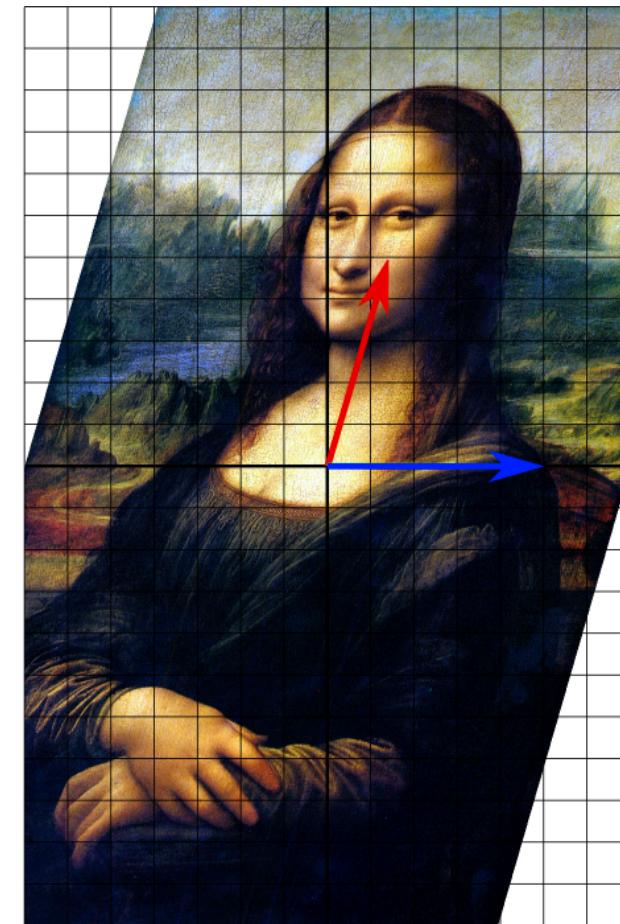
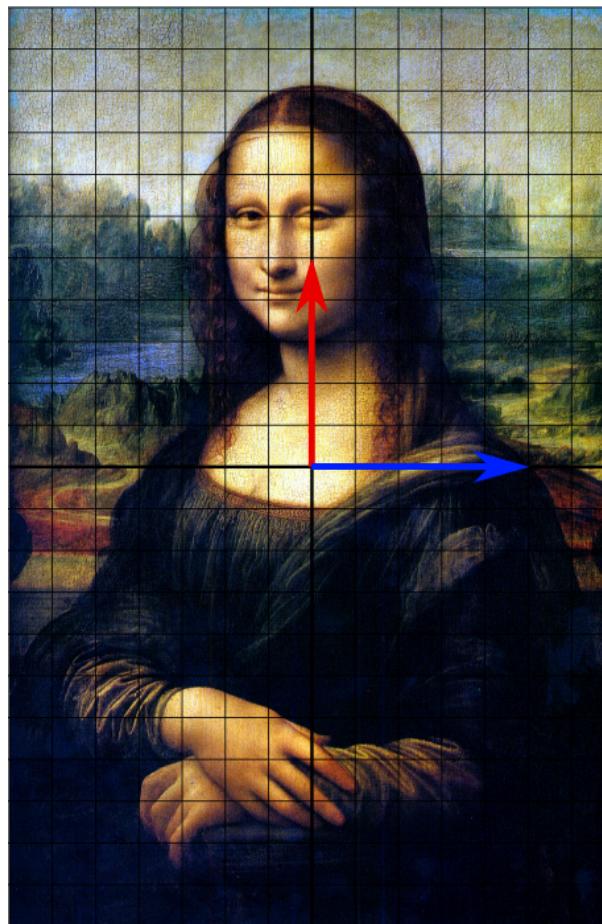


No sería brutal escribir una matriz en términos de estas matrices especiales???





Todo es cuestión de la representación que escojamos!





“Cualquier número es primo o está medido por un primo”

- Eculides, *Los Elementos*, Libro VII, Proposición 32

- Un vector no cero es autovector de una matriz cuadrada si:

$$\mathbf{Av} = \lambda \mathbf{v}$$

- El número de autovalores de una matriz viene dado por la ecuación:

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = 0 \Rightarrow \det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

- Si los autovectores correspondientes son linealmente independientes, entonces:

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}, \text{ siendo } \mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]; \mathbf{D} = \text{diag}(\lambda)$$

- Ejemplo:

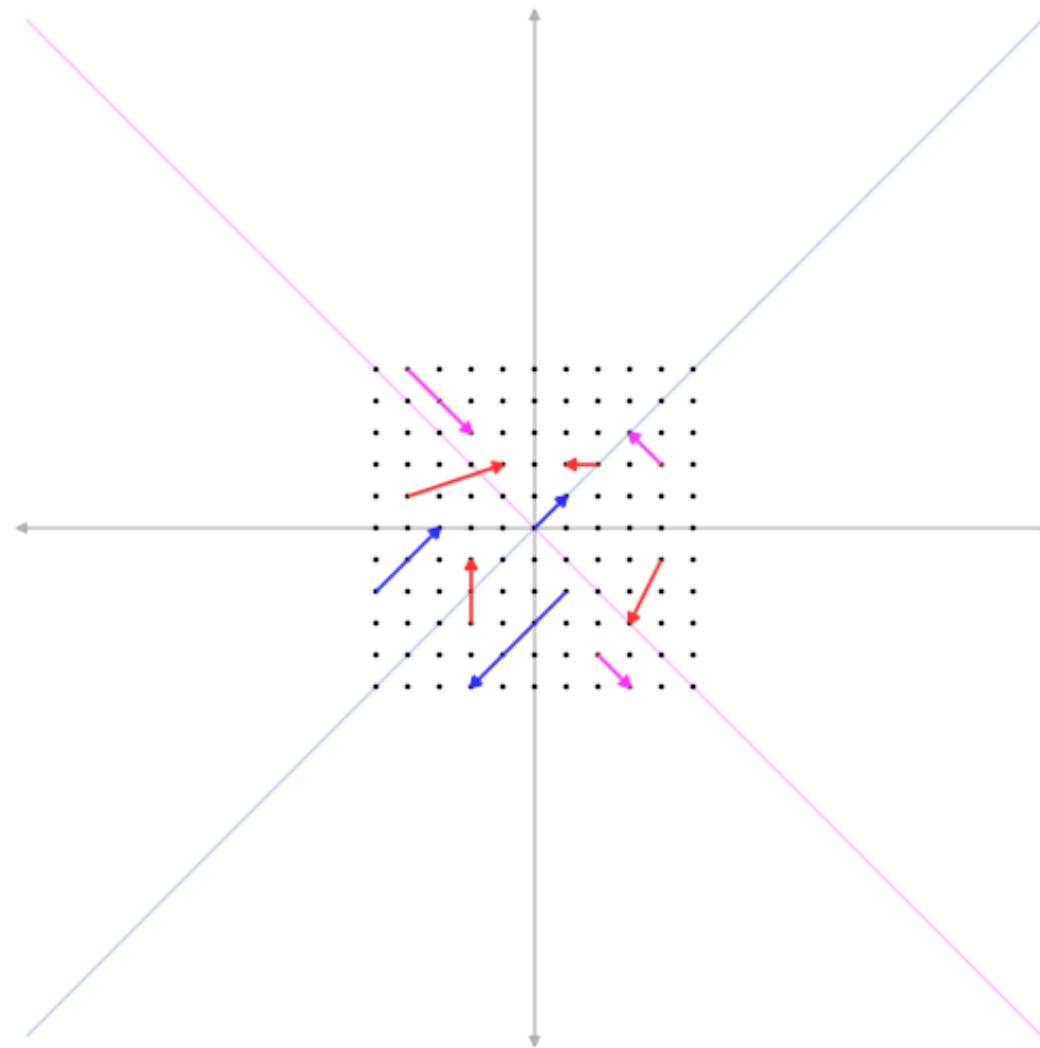
$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix}$$



“Cualquier número es primo o está medido por un primo”

- Eculides, *Los Elementos*, Libro VII, Proposición 32

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$





- Algunas propiedades importantes:

- La traza de la matriz es simplemente:

$$\text{Tr}(\mathbf{A}) = \text{Tr}(\mathbf{VDV}^{-1}) = \text{Tr}(\mathbf{DV}^{-1}\mathbf{V}) = \text{Tr}(\mathbf{D}) = \sum \lambda_i$$

- Y el determinante:

$$\det(\mathbf{A}) = \det(\mathbf{VDV}^{-1}) = \det(\mathbf{V})\det(\mathbf{D})\det(\mathbf{V}^{-1}) = \det(\mathbf{D}) = \prod \lambda_i$$

- Matriz inversa:

$$\mathbf{A}^{-1} = (\mathbf{VDV}^{-1})^{-1} = \mathbf{V}\mathbf{D}^{-1}\mathbf{V}^{-1}$$

- Si \mathbf{A} es simétrica:

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T; \text{ siendo } \mathbf{QQ}^T = \mathbf{I} \text{ y } \Lambda = \text{diag}(\lambda), \lambda \in R$$

- Si $f \in C^n$

$$f(\mathbf{A}) = \mathbf{V}f(\mathbf{D})\mathbf{V}^{-1}$$

Eigen-decomposition

Máster de Data Science

www.kschool.com





- Hasta ahora nos hemos ceñido a la descomposición (representación) de matrices cuadradas.
- El SVD (*Singular value decomposition*) es un método para factorizar cualquier matriz de números reales.
 - Fundamental en matrices no cuadradas!!!! $\mathbf{A} \in \mathbb{R}^{n \times m}$
- El PCA (*Principal Component Analysis*) es otra técnica para obtener información de una matriz. En este caso, simetrizamos el problema original, de manera que en vez de estudiar la matriz $\mathbf{A} \in \mathbb{R}^{n \times m}$, se estudia la matriz:
$$\mathbf{AA}^T \in \mathbb{R}^{n \times n}$$
- Por último, el método [Moore-Penrose psuedoinverse](#) es una forma de obtener soluciones de un sistema de ecuaciones lineales con $n \neq m$ (solución aproximada si $n < m$, o la solución con norma mínima si $n > m$).

Generalización

Máster de Data Science

www.kschool.com





- En el eigendecomposition buscábamos una factorización de una matriz cuadrada en la forma:

$$\mathbf{A} = \mathbf{V} \text{diag}(\lambda) \mathbf{V}^{-1}$$

- En particular, la aplicábamos a matrices simétricas, siendo la decomposición:

$$\mathbf{A} = \mathbf{Q} \Lambda \mathbf{Q}^T$$

- SVD se aplica a matrices $n \times m$. La factorización que buscamos es:

$$\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T$$

- \mathbf{U} es la matriz de autovectores de $\mathbf{A}\mathbf{A}^T$ (*left-singular vectors*):

$$(\mathbf{A}\mathbf{A}^T) \mathbf{U}_{:,j} = \lambda_j \mathbf{U}_{:,j}$$

- \mathbf{V} es la matriz de autovectores de $\mathbf{A}^T\mathbf{A}$ (*right-singular vectors*):

$$(\mathbf{A}^T\mathbf{A}) \mathbf{V}_{:,j} = \lambda_j \mathbf{V}_{:,j}$$

- Los valores singulares σ_i son la raíz cuadrada de los autovalores λ_i de la matriz $\mathbf{A}^T\mathbf{A}$ (iguales a los de $\mathbf{A}\mathbf{A}^T$).

$$\Sigma = \text{diag}(\sigma) = \text{diag}(\lambda)$$



- En el eigendecomposition buscábamos una factorización de una matriz cuadrada en la forma:

$$\mathbf{A} = \mathbf{V} \text{diag}(\lambda) \mathbf{V}^{-1}$$

- En particular, la aplicábamos a matrices simétricas, siendo la decomposición:

$$\mathbf{A} = \mathbf{Q} \Lambda \mathbf{Q}^T$$

- Nota: es usual utilizar una versión reducida de SVD, the *Low-Rank approximation*, donde:

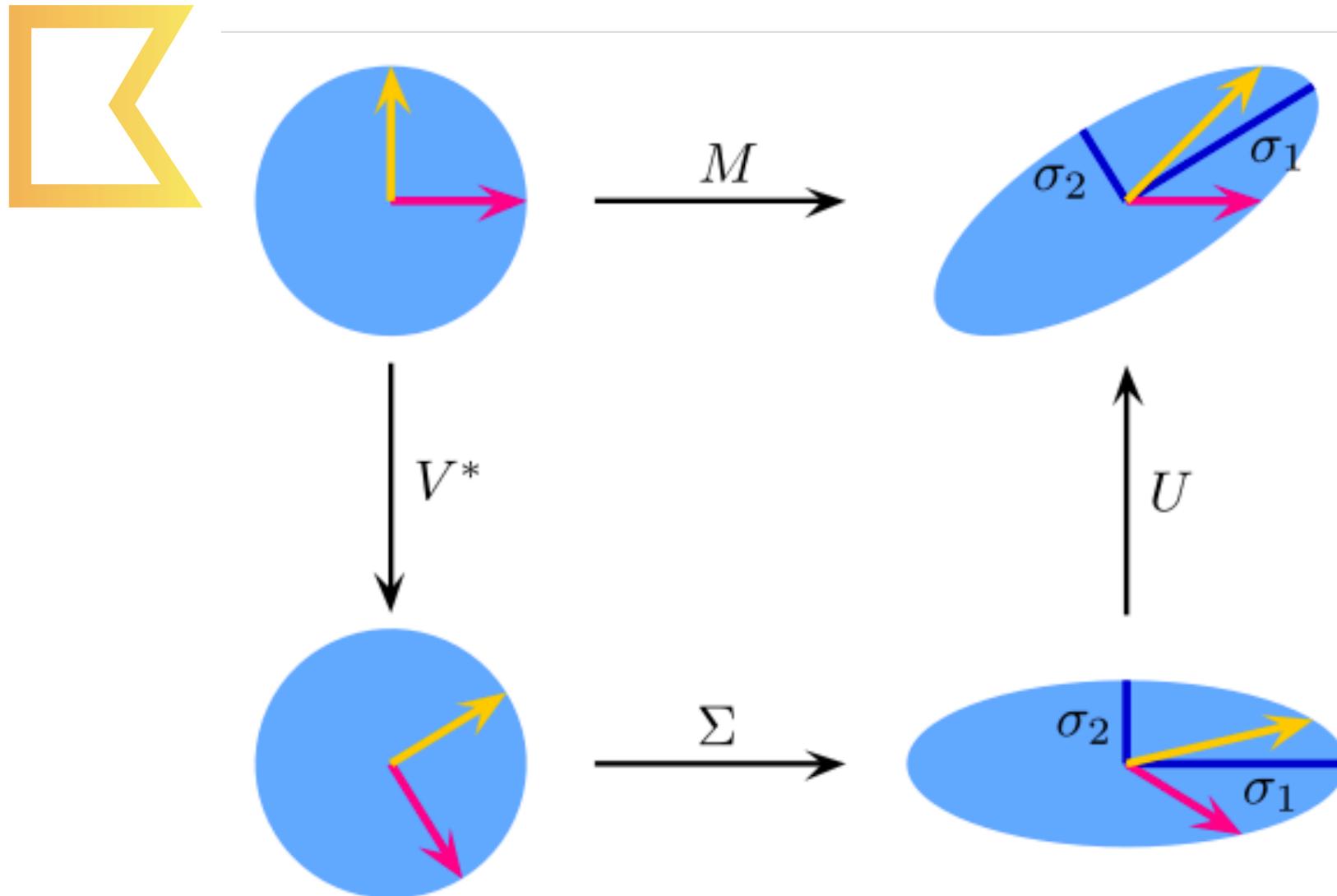
$$\mathbf{A} \in \mathbb{R}^{n \times m}; \mathbf{U} \in \mathbb{R}^{n \times r}; \Sigma \in \mathbb{R}^{r \times r}; \mathbf{V} \in \mathbb{R}^{m \times r}$$

\mathbf{U} es la matriz de autovectores de $\mathbf{A}^T \mathbf{A}$ (*right singular vectors*).

$$(\mathbf{A}^T \mathbf{A}) \mathbf{V}_{:,j} = \lambda_j \mathbf{V}_{:,j}$$

- Los valores singulares σ_i son la raíz cuadrada de los autovalores λ_i de la matriz $\mathbf{A}^T \mathbf{A}$ (iguales a los de $\mathbf{A} \mathbf{A}^T$).

$$\Sigma = \text{diag}(\sigma) = \text{diag}(\lambda)$$

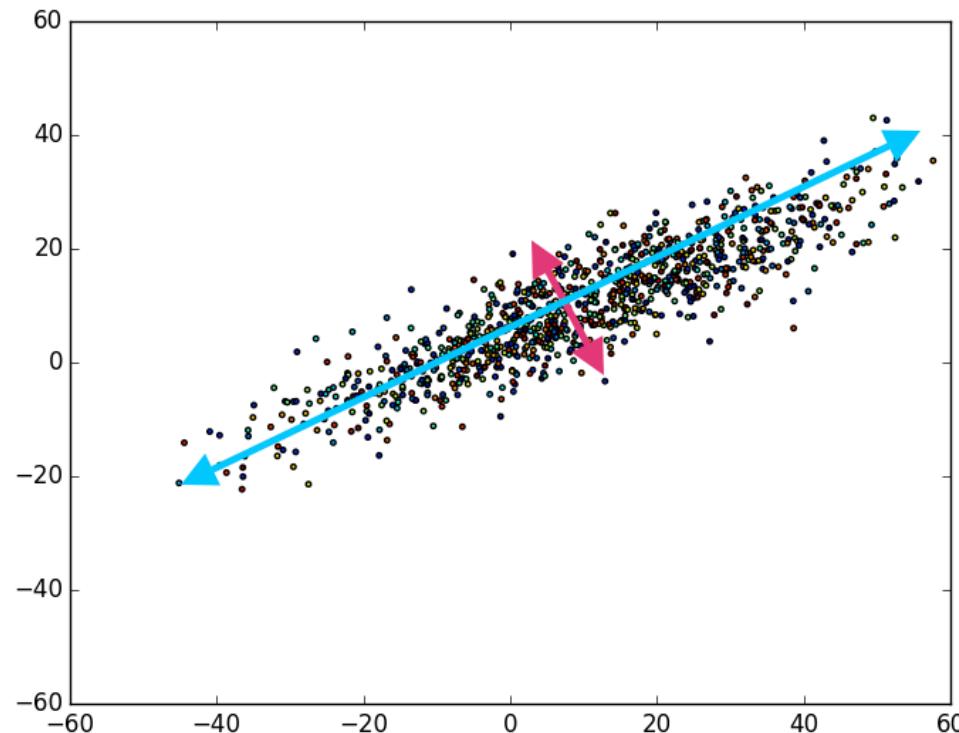


$$M = U \cdot \Sigma \cdot V^*$$

Applications of matrix decomposition

Máster de Data Science

www.kschool.com

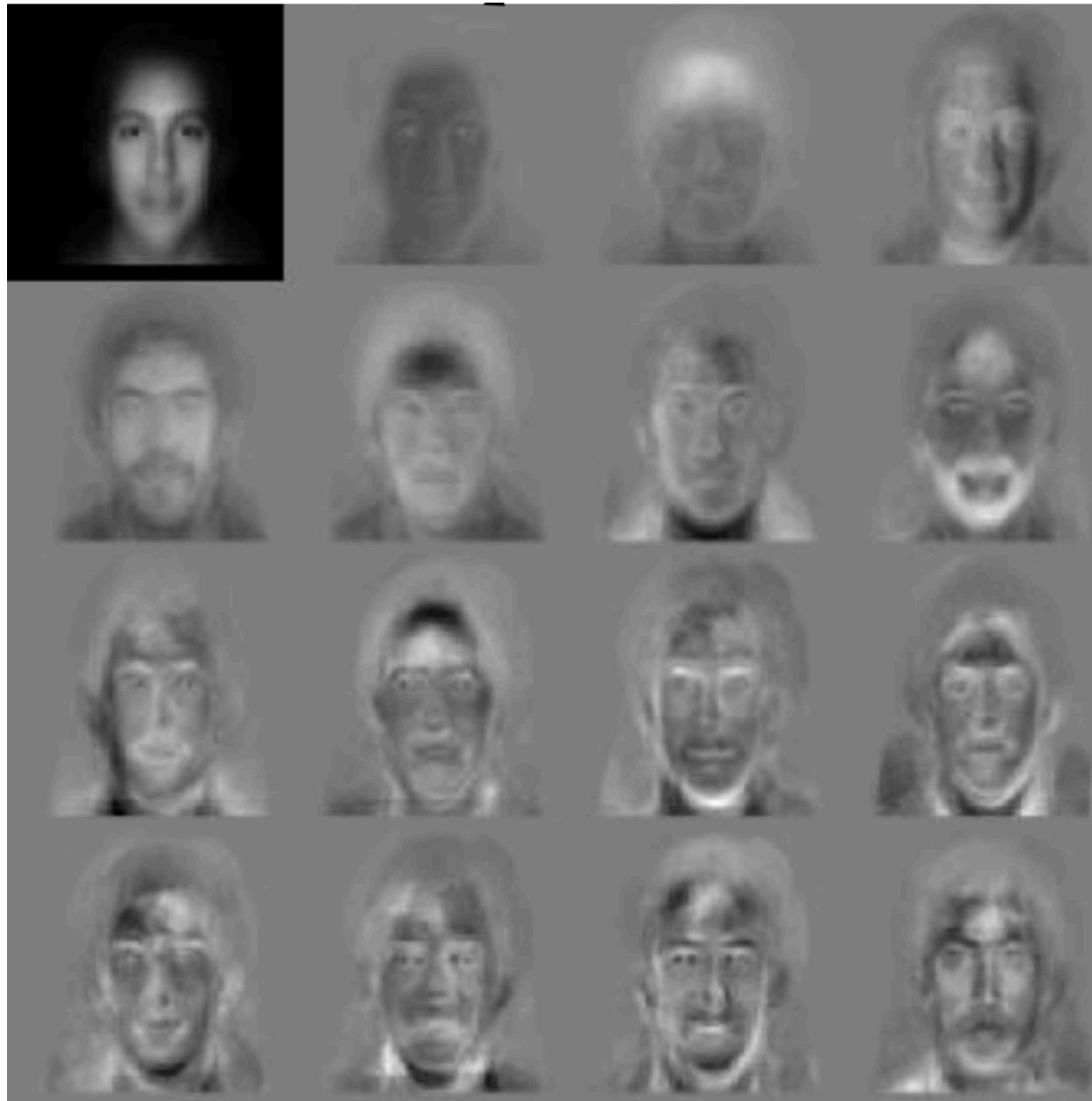


- SVD nos permite detectar las principales direcciones de simetría de nuestros problemas (*singular vectors*), y reducirlas.

Applications of matrix decomposition

Máster de Data Science

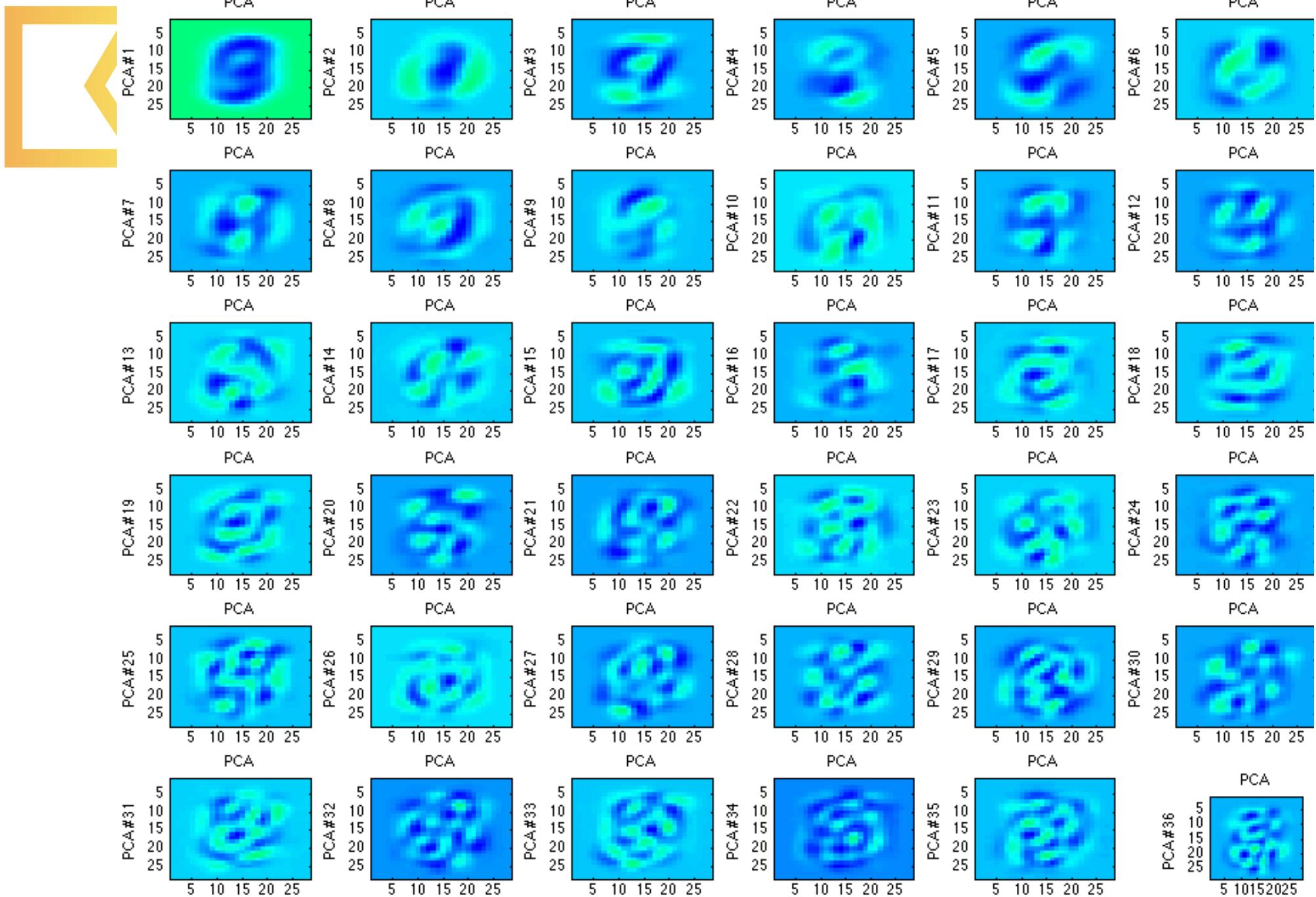
www.kschool.com



Applications of matrix decomposition

Máster de Data Science

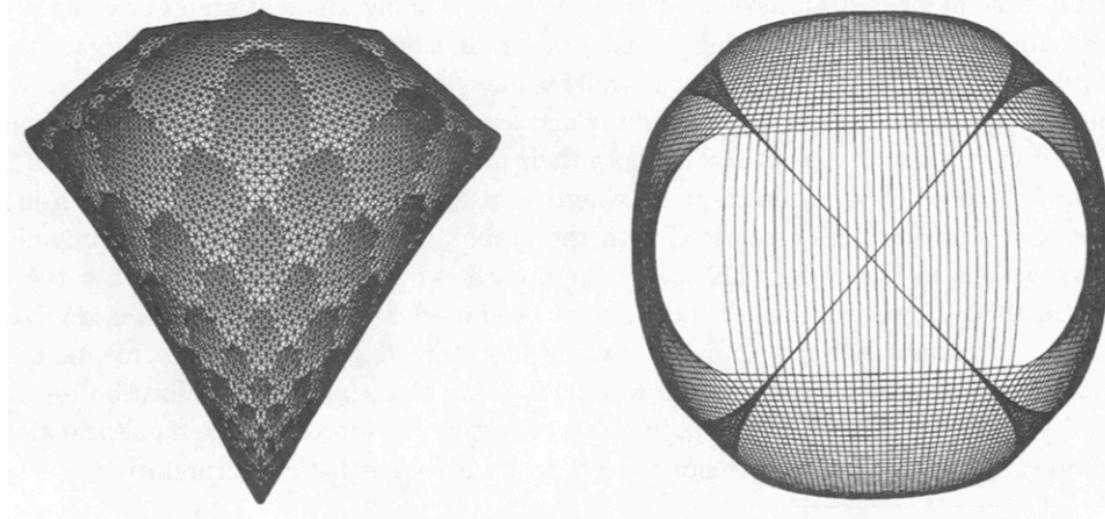
www.kschool.com



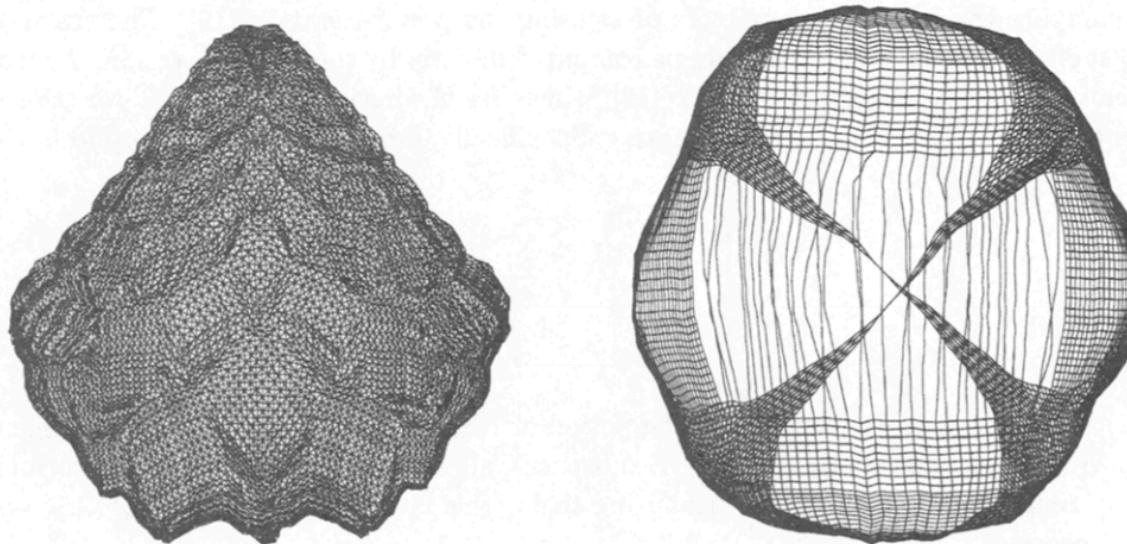
Applications of matrix decomposition

Máster de Data Science

www.kschool.com



(c) The crack graph [10]. $|V| = 10,240$, $|E| = 30,380$. (d) A 100×100 folded grid with central horizontal edge removed. $|V| = 10,000$, $|E| = 18,713$.



(c) The Crack graph [10] $|V| = 10,240$, $|E| = 30,380$ (d) A 100×100 folded grid with central horizontal edge removed. $|V| = 10,000$, $|E| = 18,713$

Applications of matrix decomposition

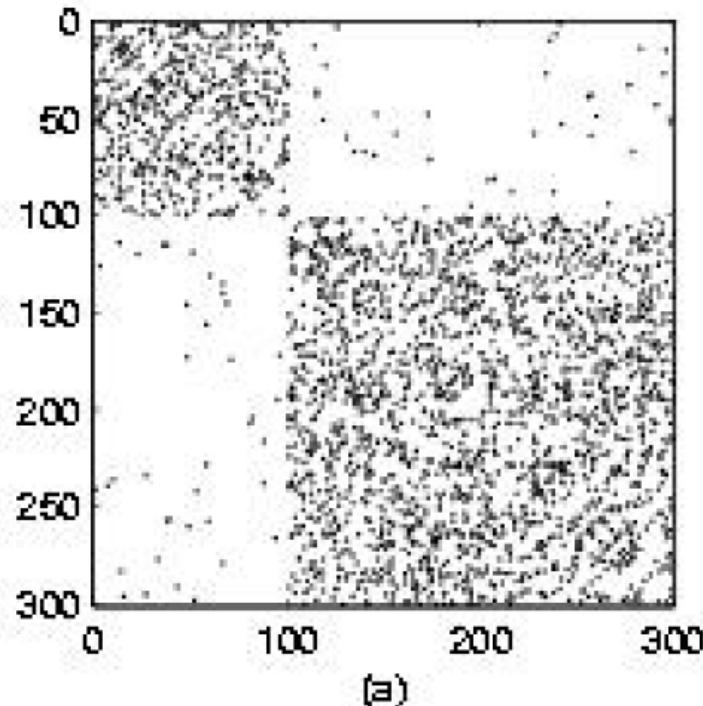
Máster de Data Science

www.kschool.com

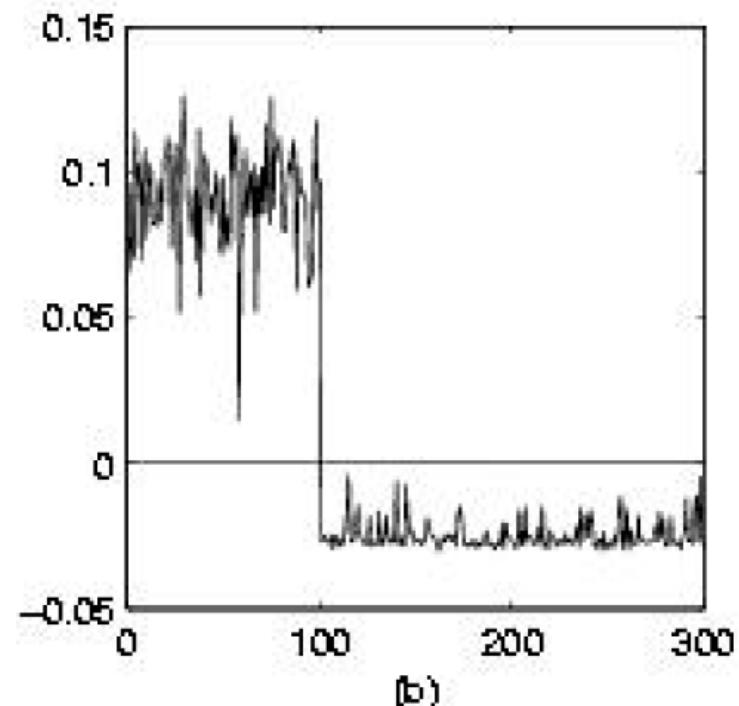


Find 2nd eigenvector of graph Laplacian (think of it as adjacency) matrix
Cluster based on 2nd eigenvector

Adjacency matrix



Eigenvector q_2



Applications of matrix decomposition

Máster de Data Science

www.k�ochschool.com

Alexander's Amazon.com > Recommended for You
(If you're not Alexander Smola, click here.)

Just For Today

[Browse Recommended](#)

Recommendations

[Amazon Instant Video](#)

[Appstore for Android](#)

[Baby](#)

[Beauty](#)

[Books](#)

[Books on Kindle](#)

[Camera & Photo](#)

[Clothing & Accessories](#)

[Computers & Accessories](#)

[Electronics](#)

[Grocery & Gourmet Food](#)

[Health & Personal Care](#)

[Home Improvement](#)

[Industrial & Scientific](#)

[Jewelry](#)

[Kitchen & Dining](#)

[MP3 Downloads](#)

[Magazine Subscriptions](#)

[Movies & TV](#)

[Music](#)

[Musical Instruments](#)

[Office & School Supplies](#)

[Patio, Lawn & Garden](#)

[Shoes](#)

[Software](#)

[Sports & Outdoors](#)

[Toys & Games](#)

[Video Games](#)

[Watches](#)

These recommendations are based on [items you own](#) and more.

view: [All](#) | [New Releases](#) | [Coming Soon](#)

[More results](#)

1. Convex Optimization



[Convex Optimization](#)

by Stephen Boyd (March 8, 2004)

Average Customer Review: (13)

In Stock

List Price: \$84.00

Price: \$68.13

[44 used & new from \\$61.32](#)

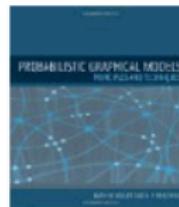
Add to Cart

Add to Wish List

I own it Not interested Rate this item

Recommended because you purchased [Nonlinear Programming](#) and more ([Fix this](#))

2. Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series)



[Probabilistic Graphical Models: Principles and Techniques \(Adaptive Computation and Machine Learning series\)](#)

by Nir Friedman (July 31, 2009)

Average Customer Review: (11)

In Stock

List Price: \$95.00

Price: \$93.55

[47 used & new from \\$91.33](#)

Add to Cart

Add to Wish List

I own it Not interested Rate this item

Recommended because you purchased [Nonlinear Programming](#) and more ([Fix this](#))

3. Doing Bayesian Data Analysis: A Tutorial with R and BUGS



[Doing Bayesian Data Analysis: A Tutorial with R and BUGS](#)

by John K. Kruschke (November 10, 2010)

Average Customer Review: (15)

In Stock

List Price: \$89.05

Price: \$77.98

[44 used & new from \\$68.40](#)

Add to Cart

Add to Wish List

I own it Not interested Rate this item

Recommended because you purchased [Bayesian Nonparametrics](#) and more ([Fix this](#))

4. Parallel and Distributed Computation: Numerical Methods (Optimization and Neural Computation)



[Parallel and Distributed Computation: Numerical Methods \(Optimization and Neural Computation\)](#)

by Dimitri P. Bertsekas (January 1, 1997)

Average Customer Review: (1)

In Stock

Price: \$49.50

[15 used & new from \\$45.49](#)

Add to Cart

Add to Wish List

I own it Not interested Rate this item

Recommended because you purchased [Nonlinear Programming](#) ([Fix this](#))

Applications of matrix decomposition

Máster de Data Science

www.kschool.com



users

1	3		5		5	4	
		5 4		4		2 1	3
2	4		1 2		3	4 3	5
	2 4		5		4		2
		4 3 4 2				2	5
1	3	3		2		4	

items

~

items

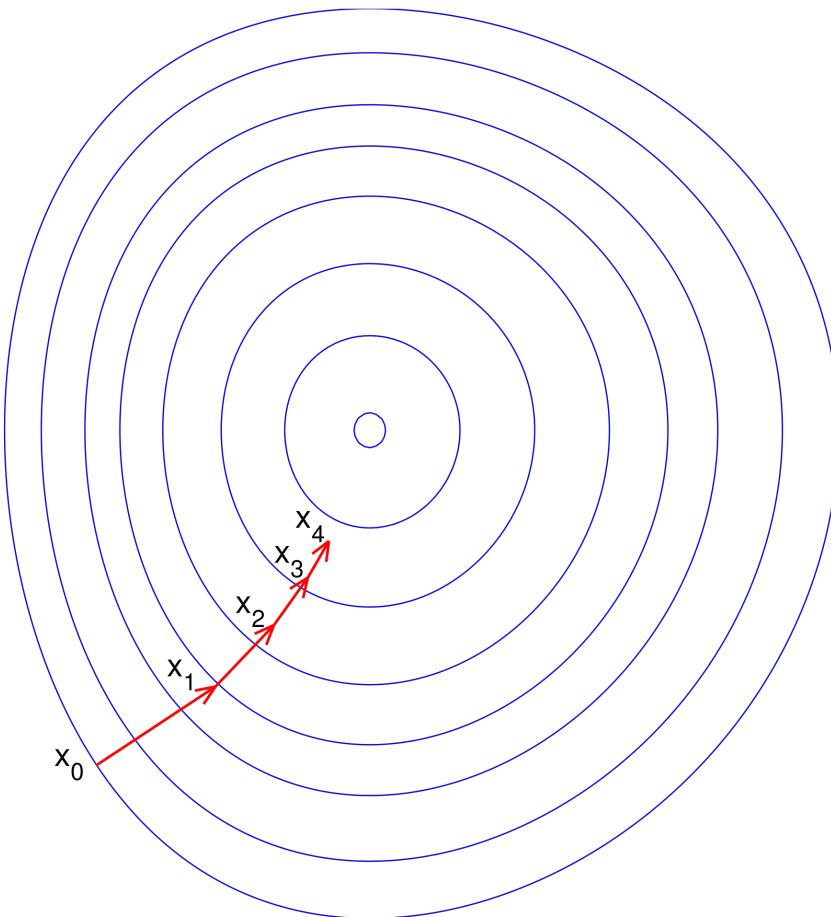
.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

~

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

A rank-3 SVD approximation



- Casi cualquier problema de **machine learning** se puede entender como una clasificación o una regresión.
- Ambos, están basados en la optimización (minimización) de un problema del tipo $\mathbf{Ax} = \mathbf{b}$.
- A la factorización de la matriz **A** (eigenvalue decomposition, SVD, PCA, LU, etc.) hay que añadirle la forma de minimizar el error en el resultado: **Gradient descent**.



- Deep learning:
 - <http://goodfeli.github.io/dlbook/>
 - <http://deeplearning.net/>
- Machine Learning:
 - Wikipedia!!!!!! Buscar supervised, unsupervised, classification, regression, collaborative filtering, gradient descent, etc.
Y recordad buscar en inglés ;)
 - Curso clásico de Coursera: <https://www.coursera.org/learn/machine-learning>
 - Libro *Mining of massive datasets*: <http://www.mmds.org/>
- Google ML library, *Tensor flow*: <https://www.tensorflow.org/>
- Una introducción a grafos que me pareció curiosa:
<http://20bits.com/article/graph-theory-part-i-introduction/>
- Otra, relacionando lo que hemos visto con Google y Facebook, muy chula ☺
<https://prezi.com/knv3fxe7lytv/graph-theory-facebook/>