Test of Time Award

Online Dictionary Learning for Sparse Coding

Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro

International Conference on Machine Learning, 2019

Test of Time Award

Online Learning for Matrix Factorization and Sparse Coding

Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro

International Conference on Machine Learning, 2019

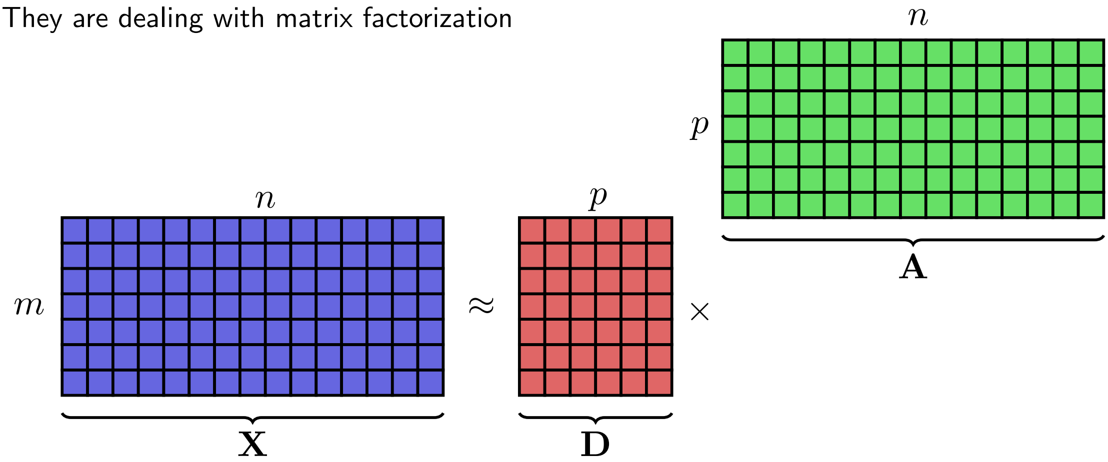
Francis

Jean

Guillermo

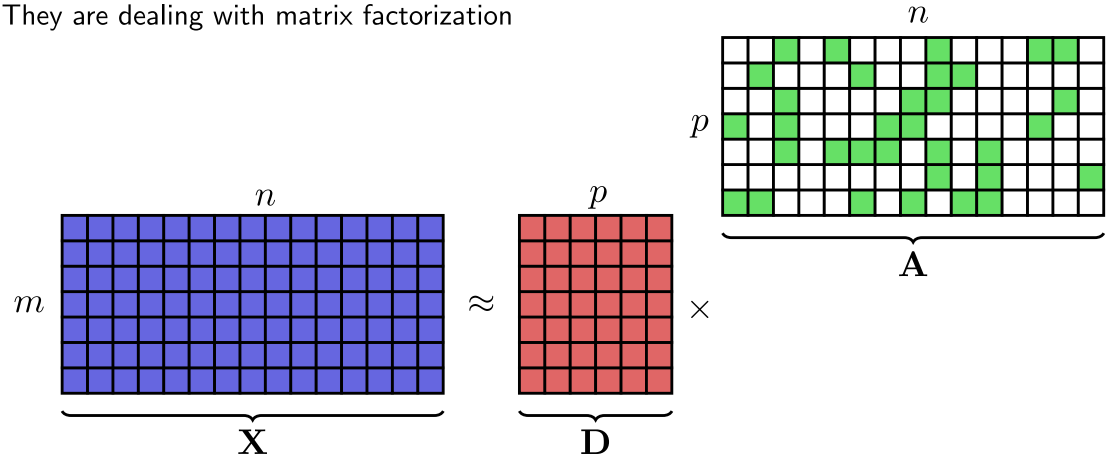
What are these papers about?

They are dealing with matrix factorization



What are these papers about?

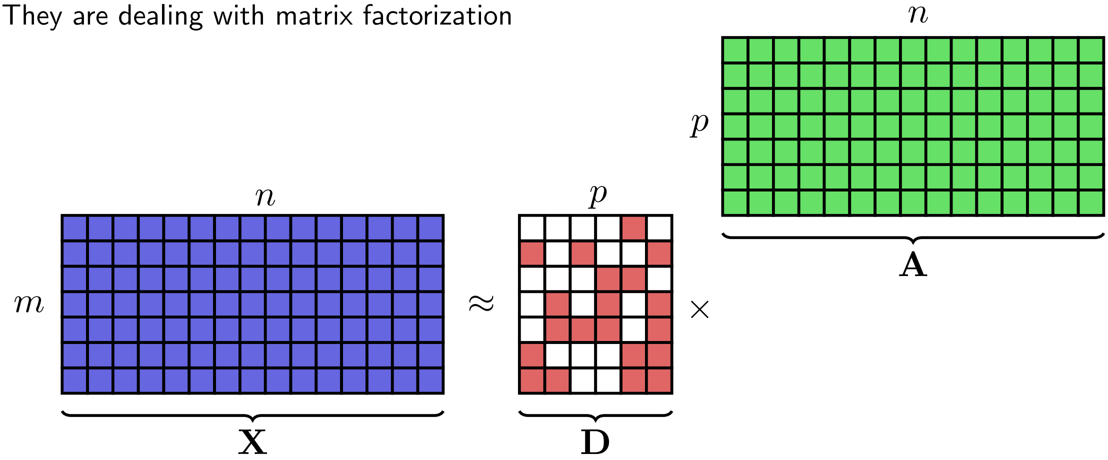
They are dealing with matrix factorization



when a factor is **sparse**.

What are these papers about?

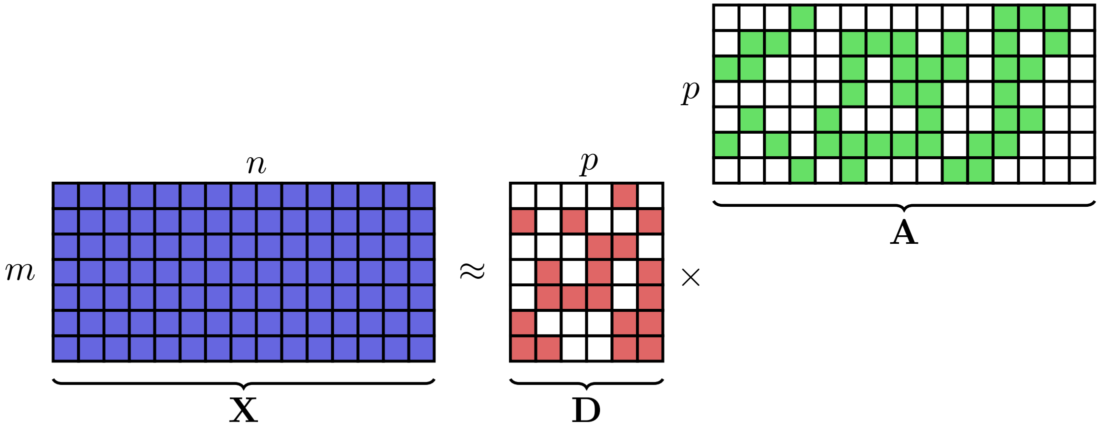
They are dealing with matrix factorization



or the other one.

What are these papers about?

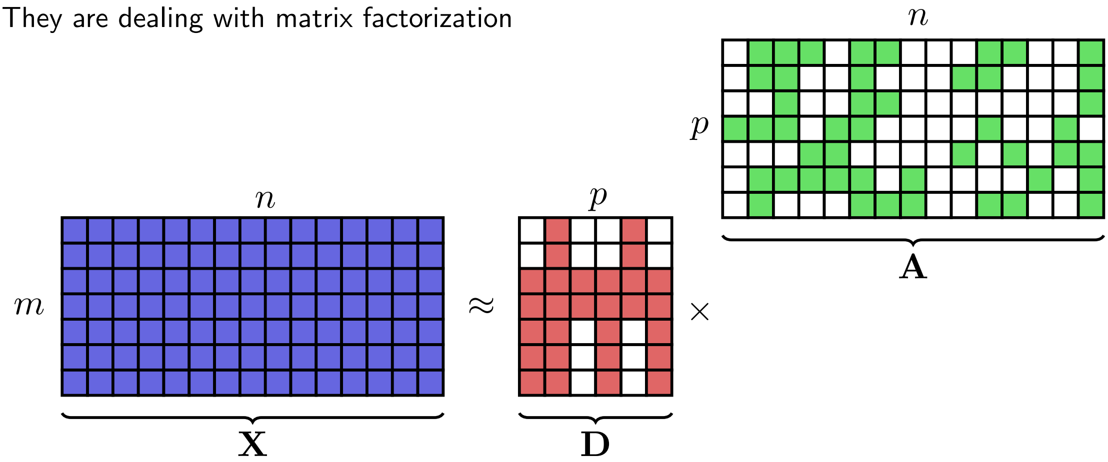
They are dealing with matrix factorization



or both.

What are these papers about?

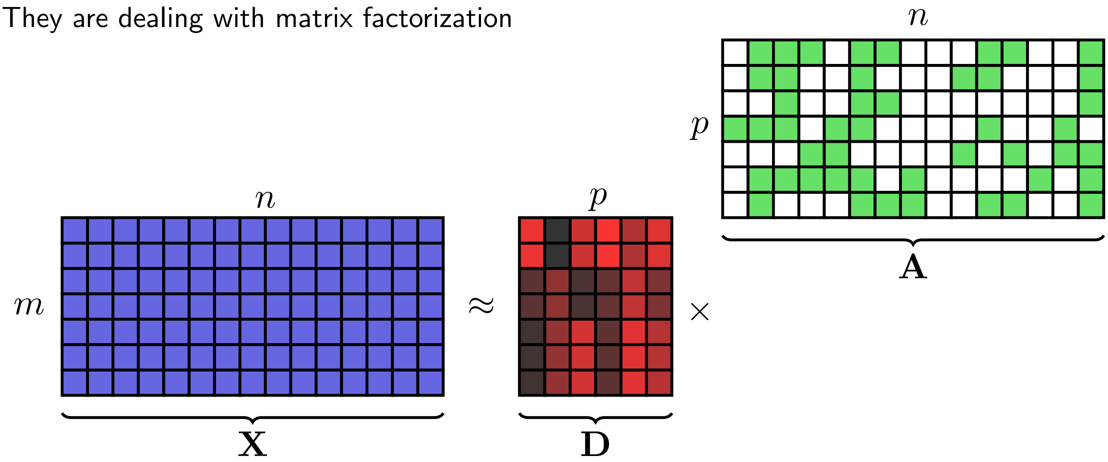
They are dealing with matrix factorization



or not only one factor is **sparse**, but it admits a **particular structure**.

What are these papers about?

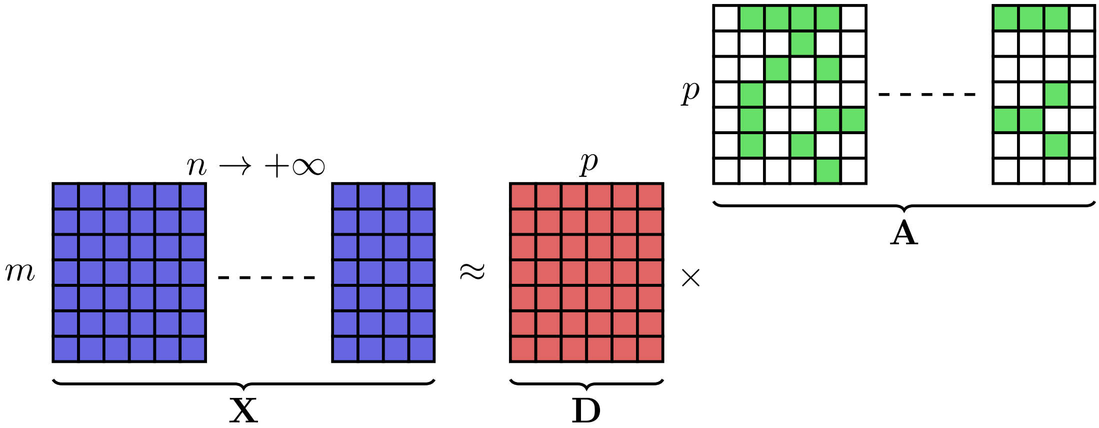
They are dealing with matrix factorization



or one factor admits a **particular structure** (*e.g.*, piecewise constant), but it is not sparse.

What these papers are about?

In these papers, data matrices have **many columns**,



or an **infinite number of columns**, or columns are **streamed online**.

Formulation(s)

- $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is a **data matrix**.
- We may call $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p]$ a **dictionary**.
- $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ carries the **decomposition coefficients** of $\mathbf{X}$ onto $\mathbf{D}$.

Formulation(s)

- $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is a **data matrix**.
- We may call $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p]$ a **dictionary**.
- $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ carries the **decomposition coefficients** of $\mathbf{X}$ onto $\mathbf{D}$.

Interpretation as signal/data decomposition

$$\mathbf{X} \approx \mathbf{DA} \quad \Longleftrightarrow \quad \forall\, i,\ \mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i = \sum_{j=1}^{p} \boldsymbol{\alpha}_i[j]\mathbf{d}_j.$$

Formulation(s)

- $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is a **data matrix**.
- We may call $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p]$ a **dictionary**.
- $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ carries the **decomposition coefficients** of $\mathbf{X}$ onto $\mathbf{D}$.

Interpretation as signal/data decomposition

$$\mathbf{X} \approx \mathbf{DA} \qquad \Longleftrightarrow \qquad \forall\ i,\ \mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i = \sum_{j=1}^{p} \boldsymbol{\alpha}_i[j]\mathbf{d}_j.$$

Generic formulation

$$\boxed{\min_{\mathbf{D} \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{x}_i, \mathbf{D}) \quad \text{with} \quad L(\mathbf{x}, \mathbf{D}) \triangleq \min_{\boldsymbol{\alpha} \in \mathcal{A}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\psi(\boldsymbol{\alpha}).}$$

Formulation(s)

- $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is a **data matrix**.
- We may call $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p]$ a **dictionary**.
- $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ carries the **decomposition coefficients** of $\mathbf{X}$ onto $\mathbf{D}$.

Interpretation as signal/data decomposition

$$\mathbf{X} \approx \mathbf{DA} \qquad \Longleftrightarrow \qquad \forall\ i,\ \mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i = \sum_{j=1}^{p} \boldsymbol{\alpha}_i[j]\mathbf{d}_j.$$

Generic formulation / stochastic case

$$\boxed{\min_{\mathbf{D} \in \mathcal{D}} \mathbb{E}_{\mathbf{x}} \left[L(\mathbf{x}, \mathbf{D}) \right] \qquad \text{with} \qquad L(\mathbf{x}, \mathbf{D}) \triangleq \min_{\boldsymbol{\alpha} \in \mathcal{A}} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\psi(\boldsymbol{\alpha}).}$$

Formulation(s)

$$\min_{\mathbf{D}\in\mathcal{D}} \mathbb{E}_{\mathbf{x}}\left[L(\mathbf{x},\mathbf{D})\right] \quad \text{with} \quad L(\mathbf{x},\mathbf{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathcal{A}} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\psi(\boldsymbol{\alpha}).$$

Which formulations does it cover?

	$\mathcal{D}$	$\mathcal{A}$	ψ
non-negative matrix factorization	$\mathbb{R}_+^{m\times p}$	$\mathbb{R}_+^p$	0

[Paatero and Tapper, '94]

Formulation(s)

$$\min_{\mathbf{D}\in\mathcal{D}} \mathbb{E}_{\mathbf{x}}\left[L(\mathbf{x},\mathbf{D})\right] \quad \text{with} \quad L(\mathbf{x},\mathbf{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathcal{A}} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\psi(\boldsymbol{\alpha}).$$

Which formulations does it cover?

	$\mathcal{D}$	$\mathcal{A}$	ψ
non-negative matrix factorization	$\mathbb{R}_+^{m\times p}$	$\mathbb{R}_+^p$	0
sparse coding	$\{\mathbf{D} : \forall\ j,\ \ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1$

[Paatero and Tapper, '94], [Olshausen and Field, '96]

Formulation(s)

$$\min_{\mathbf{D}\in\mathcal{D}} \mathbb{E}_{\mathbf{x}}\left[L(\mathbf{x},\mathbf{D})\right] \quad \text{with} \quad L(\mathbf{x},\mathbf{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathcal{A}} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\psi(\boldsymbol{\alpha}).$$

Which formulations does it cover?

	$\mathcal{D}$	$\mathcal{A}$	ψ
non-negative matrix factorization	$\mathbb{R}_+^{m\times p}$	$\mathbb{R}_+^p$	0
sparse coding	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1$
non-negative sparse coding	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}_+^p$	$\|.\|_1$

[Paatero and Tapper, '94], [Olshausen and Field, '96], [Hoyer, 2002]

Formulation(s)

$$\min_{\mathbf{D}\in\mathcal{D}} \mathbb{E}_{\mathbf{x}}\left[L(\mathbf{x},\mathbf{D})\right] \quad \text{with} \quad L(\mathbf{x},\mathbf{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathcal{A}} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\psi(\boldsymbol{\alpha}).$$

Which formulations does it cover?

	$\mathcal{D}$	$\mathcal{A}$	ψ
non-negative matrix factorization	$\mathbb{R}_+^{m\times p}$	$\mathbb{R}_+^p$	0
sparse coding	$\{\mathbf{D} : \forall\ j,\ \ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1$
non-negative sparse coding	$\{\mathbf{D} : \forall\ j,\ \ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}_+^p$	$\|.\|_1$
structured sparse coding	$\{\mathbf{D} : \forall\ j,\ \ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1 + \Omega(.)$

[Paatero and Tapper, '94], [Olshausen and Field, '96], [Hoyer, 2002], [Mairal et al., 2011]

Formulation(s)

$$\min_{\mathbf{D}\in\mathcal{D}} \mathbb{E}_{\mathbf{x}}\left[L(\mathbf{x},\mathbf{D})\right] \quad \text{with} \quad L(\mathbf{x},\mathbf{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathcal{A}} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\psi(\boldsymbol{\alpha}).$$

Which formulations does it cover?

	$\mathcal{D}$	$\mathcal{A}$	ψ
non-negative matrix factorization	$\mathbb{R}_+^{m\times p}$	$\mathbb{R}_+^p$	0
sparse coding	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1$
non-negative sparse coding	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}_+^p$	$\|.\|_1$
structured sparse coding	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1 + \Omega(.)$
$\approx$ **sparse PCA**	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\|_2^2 + \|\mathbf{d}_j\|_1 \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1$

[Paatero and Tapper, '94], [Olshausen and Field, '96], [Hoyer, 2002], [Mairal et al., 2011], [Zou et al., 2004].

Formulation(s)

$$\min_{\mathbf{D}\in\mathcal{D}} \mathbb{E}_{\mathbf{x}}\left[L(\mathbf{x},\mathbf{D})\right] \quad \text{with} \quad L(\mathbf{x},\mathbf{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathcal{A}} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\psi(\boldsymbol{\alpha}).$$

Which formulations does it cover?

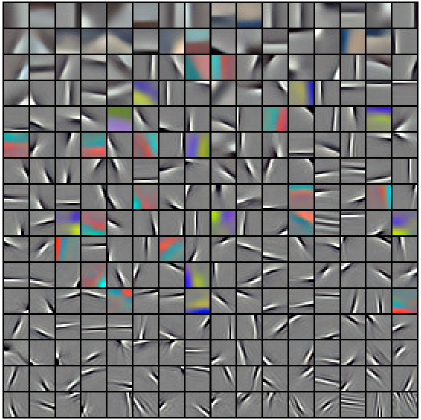
	$\mathcal{D}$	$\mathcal{A}$	ψ
non-negative matrix factorization	$\mathbb{R}_+^{m\times p}$	$\mathbb{R}_+^p$	0
sparse coding	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1$
non-negative sparse coding	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}_+^p$	$\|.\|_1$
structured sparse coding	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\| \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1 + \Omega(.)$
$\approx$ **sparse PCA**	$\{\mathbf{D} : \forall\, j,\ \|\mathbf{d}_j\|_2^2 + \|\mathbf{d}_j\|_1 \leq 1\}$	$\mathbb{R}^p$	$\|.\|_1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

[Paatero and Tapper, '94], [Olshausen and Field, '96], [Hoyer, 2002], [Mairal et al., 2011], [Zou et al., 2004].

The sparse coding context

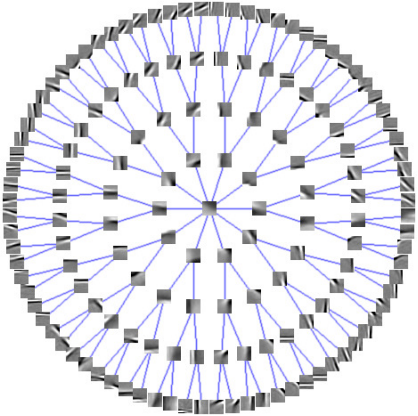
was introduced by Olshausen and Field, '96. It was the first time (together with ICA, see [Bell and Sejnowski, '97]) that a **simple unsupervised learning principle** would lead to

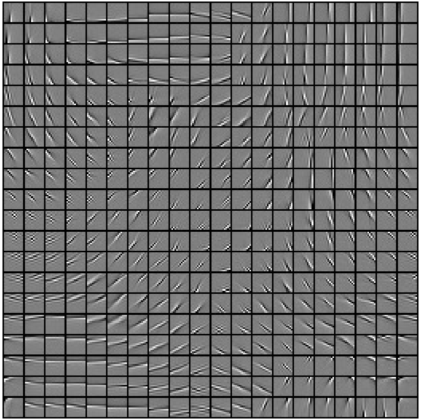
various sorts of "Gabor-like" filters, when trained on natural image patches.



The sparse coding context

Remember that we can play with various **structured sparsity-inducing penalties**:





[Jenatton et al. 2010], [Kavukcuoglu et al., 2009], [Mairal et al. 2011], [Hyvärinen and Hoyer, 2001].

Sparsity and simplicity principles

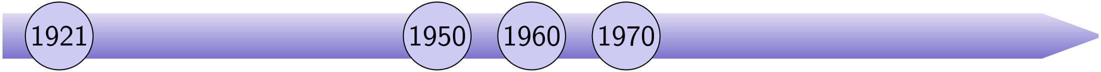
1921

- 1921: Wrinch and Jeffrey's simplicity principle.

Sparsity and simplicity principles

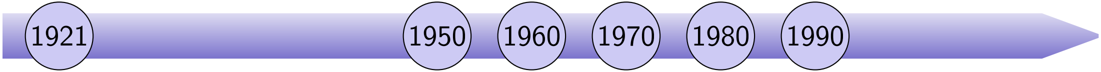
- 1921: Wrinch and Jeffrey's simplicity principle.
- 1952: Markowitz's portfolio selection.

Sparsity and simplicity principles



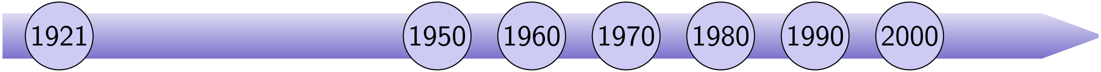
- 1921: Wrinch and Jeffrey's simplicity principle.
- 1952: Markowitz's portfolio selection.
- 1960's and 70's: best subset selection in statistics.

Sparsity and simplicity principles



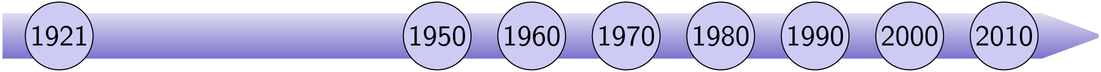
- 1921: Wrinch and Jeffrey's simplicity principle.
- 1952: Markowitz's portfolio selection.
- 1960's and 70's: best subset selection in statistics.
- 1990's: the **wavelet era** in signal processing.

Sparsity and simplicity principles



- 1921: Wrinch and Jeffrey's simplicity principle.
- 1952: Markowitz's portfolio selection.
- 1960's and 70's: best subset selection in statistics.
- 1990's: the **wavelet era** in signal processing.
- 1996: Olshausen and Field's **dictionary learning** method.
- 1994–1996: the **Lasso** (Tibshirani) and **Basis pursuit** (Chen and Donoho).

Sparsity and simplicity principles



- 1921: Wrinch and Jeffrey's simplicity principle.
- 1952: Markowitz's portfolio selection.
- 1960's and 70's: best subset selection in statistics.
- 1990's: the **wavelet era** in signal processing.
- 1996: Olshausen and Field's **dictionary learning** method.
- 1994–1996: the **Lasso** (Tibshirani) and **Basis pursuit** (Chen and Donoho).
- 2004: **compressed sensing** (Candes, Romberg and Tao).
- 2006: Elad and Aharon's image denoising method.

Context of 2009

Context of 2009

Many successful stories of dictionary learning in image processing

- image denoising, inpainting, demosaicing, super-resolution ...

[Elad and Aharon., 2006], [Mairal et al., 2008], [Yang et al., 2008] ...

Context of 2009

Many successful stories of dictionary learning in image processing

- image denoising, inpainting, demosaicing, super-resolution . . .

[Elad and Aharon., 2006], [Mairal et al., 2008], [Yang et al., 2008] . . .

Context of 2009

Many successful stories of dictionary learning in image processing

- image denoising, inpainting, demosaicing, super-resolution ...

Also successful stories in computer vision for modeling local features

- dictionary learning on top of SIFT wins the PASCAL VOC'09 challenge.
- another variant wins the ImageNet 2010 challenge.

[Yang et al., 2009], [Lin et al. 2010] ...

Context of 2009

Many successful stories of dictionary learning in image processing

- image denoising, inpainting, demosaicing, super-resolution ...

Also successful stories in computer vision for modeling local features

- dictionary learning on top of SIFT wins the PASCAL VOC'09 challenge.
- another variant wins the ImageNet 2010 challenge.

Matrix factorization becomes a key technique for unsupervised data modeling

- recommender systems (Netflix prize) and social networks.
- document clustering.
- genomic pattern discovery.
- ...

[Koren et al., 2009b], [Ma et al. 2008], [Xu et al. 2003], [Brunet et al., 2004]...

Context of 2009

Classical approach for matrix factorization: alternate minimization

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}\in\mathcal{A}} \frac{1}{2}\|\mathbf{X}-\mathbf{D}\mathbf{A}\|_{\mathsf{F}}^2 + \lambda\psi(\mathbf{A}).$$

which requires loading all data at every iteration (**batch optimization**).

Context of 2009

Classical approach for matrix factorization: alternate minimization

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}\in\mathcal{A}} \frac{1}{2}\|\mathbf{X}-\mathbf{D}\mathbf{A}\|_{\mathsf{F}}^2 + \lambda\psi(\mathbf{A}).$$

which requires loading all data at every iteration (**batch optimization**).

Meanwhile, Léon Bottou is advocating stochastic optimization for machine learning

see Léon's tutorial at NIPS'07, or NeurIPS'18 test of time award [Bottou and Bousquet, 2008].

Context of 2009

Classical approach for matrix factorization: alternate minimization

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}\in\mathcal{A}} \frac{1}{2}\|\mathbf{X}-\mathbf{D}\mathbf{A}\|_{\mathsf{F}}^2 + \lambda\psi(\mathbf{A}).$$

which requires loading all data at every iteration (**batch optimization**).

Meanwhile, Léon Bottou is advocating stochastic optimization for machine learning

which makes the risk minimization point of view relevant:

$$\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{x}_i,\mathbf{D}) \quad \text{with} \quad L(\mathbf{x},\mathbf{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathcal{A}} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}\|^2+\lambda\psi(\boldsymbol{\alpha}).$$

see Léon's tutorial at NIPS'07, or NeurIPS'18 test of time award [Bottou and Bousquet, 2008].

What we did

We started experimenting with SGD

What we did

We started experimenting with SGD and tuning the step-size turned out to be painful.

What we did

We started experimenting with SGD and tuning the step-size turned out to be painful.

Can we then design an algorithm that would be as fast as SGD, but more practical?

What we did

We started experimenting with SGD and tuning the step-size turned out to be painful.

Can we then design an algorithm that would be as fast as SGD, but more practical?

- **Idea 1**: If we knew optimal codes $\boldsymbol{\alpha}_i^\star$ for all $\mathbf{x}_i$'s in advance, then the problem becomes

$$\min_{\mathbf{D}\in\mathcal{D}} \left\{ \frac{1}{2}\text{trace}\left(\mathbf{D}^\top\mathbf{D}\mathbf{B}\right) - \text{trace}\left(\mathbf{D}^\top\mathbf{C}\right) \right\} \text{with } \mathbf{B} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\alpha}_i^\star\boldsymbol{\alpha}_i^{\star\top} \text{ and } \mathbf{C} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\boldsymbol{\alpha}_i^{\star\top},$$

which yields **parameter-free** block coordinate descent rules for updating $\mathbf{D}$.

What we did

We started experimenting with SGD and tuning the step-size turned out to be painful.

Can we then design an algorithm that would be as fast as SGD, but more practical?

- **Idea 1**: If we knew optimal codes $\boldsymbol{\alpha}_i^\star$ for all $\mathbf{x}_i$'s in advance, then the problem becomes

$$\min_{\mathbf{D}\in\mathcal{D}} \left\{ \frac{1}{2}\text{trace}\left(\mathbf{D}^\top\mathbf{D}\mathbf{B}\right) - \text{trace}\left(\mathbf{D}^\top\mathbf{C}\right)\right\} \text{with } \mathbf{B} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\alpha}_i^\star\boldsymbol{\alpha}_i^{\star\top} \text{ and } \mathbf{C} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\boldsymbol{\alpha}_i^{\star\top},$$

  which yields **parameter-free** block coordinate descent rules for updating $\mathbf{D}$.
- **Idea 2**: Build appropriate matrices $\mathbf{B}$ and $\mathbf{C}$ **in an online fashion**.

[Neal and Hinton, '98]

What we did

We started experimenting with SGD and tuning the step-size turned out to be painful.

Can we then design an algorithm that would be as fast as SGD, but more practical?

- **Idea 1**: If we knew optimal codes $\boldsymbol{\alpha}_i^\star$ for all $\mathbf{x}_i$'s in advance, then the problem becomes

$$\min_{\mathbf{D} \in \mathcal{D}} \left\{ \frac{1}{2} \mathrm{trace}\left(\mathbf{D}^\top \mathbf{D} \mathbf{B}\right) - \mathrm{trace}\left(\mathbf{D}^\top \mathbf{C}\right) \right\} \text{with } \mathbf{B} = \frac{1}{n} \sum_{i=1}^n \boldsymbol{\alpha}_i^\star \boldsymbol{\alpha}_i^{\star\top} \text{ and } \mathbf{C} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \boldsymbol{\alpha}_i^{\star\top},$$

  which yields **parameter-free** block coordinate descent rules for updating $\mathbf{D}$.
- **Idea 2**: Build appropriate matrices $\mathbf{B}$ and $\mathbf{C}$ **in an online fashion**.

What about theory?

We could provide guarantees of convergence to stationary points, even though the problem is **non-convex**, **stochastic**, **constrained**, and **non-smooth**.

[Neal and Hinton, '98], [Mairal, 2013], [Mensch, 2018].

Reasons for impact: How did it help other fields?

Reasons for impact: How did it help other fields?

A timely context ($\approx$ luck)

Datasets were becoming larger and larger, and there was **suddenly a need for more scalable** matrix factorization methods.

Reasons for impact: How did it help other fields?

A timely context ($\approx$ luck)

Datasets were becoming larger and larger, and there was **suddenly a need for more scalable** matrix factorization methods.

A combination of mathematics and engineering?

- **an efficient software package**: the SPAMS toolbox.
- **robustness to hyper-parameters:** default setting that works (many times) in practice.
- (try it with `pip install spams` in Python, or download R/Matlab packages).

Reasons for impact: How did it help other fields?

A timely context ($\approx$ luck)

Datasets were becoming larger and larger, and there was **suddenly a need for more scalable** matrix factorization methods.

A combination of mathematics and engineering?

- **an efficient software package**: the SPAMS toolbox.
- **robustness to hyper-parameters:** default setting that works (many times) in practice.
- (try it with `pip install spams` in Python, or download R/Matlab packages).

Flexibility in the constraints/penalty design

- **allowing the method to be used in unexpected contexts.**

Reasons for impact: How did it help other fields?

A timely context ($\approx$ luck)

Datasets were becoming larger and larger, and there was **suddenly a need for more scalable** matrix factorization methods.

A combination of mathematics and engineering?

- **an efficient software package**: the SPAMS toolbox.
- **robustness to hyper-parameters:** default setting that works (many times) in practice.

PNAS

Computer vision cracks the leaf code

Peter Wilf[a,1], Shengping Zhang[b,c,1], Sharat Chikkerur[d], Stefan A. Little[a,e], Scott L. Wing[f], and Thomas Serre[b,1]

[a]Department of Geosciences, Pennsylvania State University, University Park, PA 16802; [b]Department of Cognitive, Linguistic and Psychological Sciences, Brown Institute for Brain Science, Brown University, Providence, RI 02912; [c]School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, Shandong, People's Republic of China; [d]Azure Machine Learning, Microsoft, Cambridge, MA 02142; [e]Laboratoire Ecologie, Systématique et Evolution, Université Paris-Sud, 91405 Orsay Cedex, France; and [f]Department of Paleobiology, National Museum of Natural History, Smithsonian Institution, Washington, DC 20013

Edited by Andrew H. Knoll, Harvard University, Cambridge, MA, and approved February 1, 2016 (received for review December 14, 2015)

Understanding the extremely variable, complex shape and venation characters of angiosperm leaves is one of the most challenging

species (15–19), and there is community interest in approaching this problem through crowd-sourcing of images and machine-identifi-

Reasons for impact: How did it help other fields?

A timely context ($\approx$ luck)

Datasets were becoming larger and larger, and there was **suddenly a need for more scalable** matrix factorization methods.

A combination of mathematics and engineering?

- **an efficient software package**: the SPAMS toolbox.
- **robustness to hyper-parameters:** default setting that works (many times) in practice.

PNAS

Stability-driven nonnegative matrix factorization to interpret spatial gene expression and build local gene networks

Siqi Wu[a,b], Antony Joseph[a,b,c], Ann S. Hammonds[b], Susan E. Celniker[b], Bin Yu[a,d,1], and Erwin Frise[b,1]

[a]Department of Statistics, University of California, Berkeley, CA 94720; [b]Division of Environmental Genomics and Systems Biology, Lawrence Berkeley National Laboratory, Berkeley, CA 94720; [c]Walmart Labs, San Bruno, CA 94066; and [d]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720

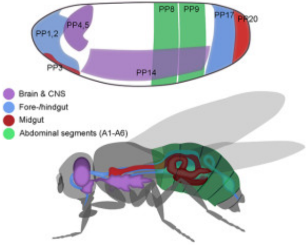
Reasons for impact: How did it help other fields?

A timely context ($\approx$ luck)

Datasets were becoming larger and larger, and there was **suddenly a need for more scalable** matrix factorization methods.

A combination of mathematics and engineering?

- **an efficient software package**: the SPAMS toolbox.
- **robustness to hyper-parameters:** default setting that works (many times) in practice.
- (try it with `pip install spams` in Python, or download R/M



Flexibility in the constraints/penalty design

Mapping a Cell's Destiny

New Berkeley Lab Tool Speeds Discovery of Spatial Patterns in Gene Networks

Science Shorts Sarah Yang (510) 486-4575 • MAY 4, 2016

Reasons for impact: How did it help other fields?

A timely context ($\approx$ luck)

Datasets were becoming larger and larger, and there was **suddenly a need for more scalable** matrix factorization methods.

A combination of mathematics and engineering?

- **an e**
- **robu** ... practice.
- (try ...

Flexibilit

- **allo**

Correcting gene expression data when neither the unwanted variation nor the factor of interest are observed

LAURENT JACOB*

Laboratoire de Biométrie et Biologie Évolutive, Université de Lyon, Université Lyon 1, CNRS, UMR, 5558 Lyon, France

laurent.jacob@univ-lyon1.fr

JOHANN A. GAGNON-BARTSCH

Department of Statistics, University of California, Berkeley, CA 974720, USA

TERENCE P. SPEED

Department of Statistics, University of California, Berkeley, CA 974720, USA and Division of

Connection with neural networks

Connection with neural networks

A cheap way to obtain a sparse code $\boldsymbol{\beta}$ from $\mathbf{x}$ and $\mathbf{D}$ is

$$\boldsymbol{\beta} = \mathsf{relu}(\mathbf{D}^\top \mathbf{x} - \lambda),$$

versus

$$\boldsymbol{\alpha} \in \arg\min_{\boldsymbol{\alpha} \in \mathcal{A}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1.$$

Connection with neural networks

A cheap way to obtain a sparse code $\boldsymbol{\beta}$ from $\mathbf{x}$ and $\mathbf{D}$ is

$$\boldsymbol{\beta} = \mathsf{relu}(\mathbf{D}^\top \mathbf{x} - \lambda),$$

versus

$$\boldsymbol{\alpha} \in \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \mathcal{A}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1.$$

Then, not surprisingly, for dictionary learning,

- **end-to-end feature learning** is feasible.

[Mairal et al., 2012]

Connection with neural networks

A cheap way to obtain a sparse code $\boldsymbol{\beta}$ from $\mathbf{x}$ and $\mathbf{D}$ is

$$\boldsymbol{\beta} = \mathsf{relu}(\mathbf{D}^\top \mathbf{x} - \lambda),$$

versus

$$\boldsymbol{\alpha} \in \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \mathcal{A}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1.$$

Then, not surprisingly, for dictionary learning,

- **end-to-end feature learning** is feasible.
- one can design **convolutional** and **multilayer** models.

[Mairal et al., 2012], [Zeiler and Fergus, 2010]

Connection with neural networks

A cheap way to obtain a sparse code $\boldsymbol{\beta}$ from $\mathbf{x}$ and $\mathbf{D}$ is

$$\boldsymbol{\beta} = \mathsf{relu}(\mathbf{D}^\top \mathbf{x} - \lambda),$$

versus

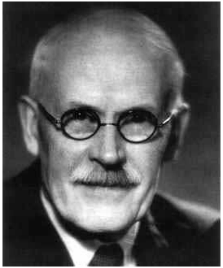
$$\boldsymbol{\alpha} \in \arg\min_{\boldsymbol{\alpha} \in \mathcal{A}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1.$$

Then, not surprisingly, for dictionary learning,

- **end-to-end feature learning** is feasible.
- one can design **convolutional** and **multilayer** models.
- sparse decomposition algorithms perform **neural network-like** operations (LISTA).

[Mairal et al., 2012], [Zeiler and Fergus, 2010], [Gregor and LeCun, 2010].

Thoughts



Is Wrinch and Jeffrey's simplicity principle still relevant?

- Simplicity is a key to **interpretability** and to **model/hypothesis selection**.
- Next form will probably be different than ℓ_1. Which one?
- Simplicity is not enough. Various forms and **robustness and stability** are also needed.

[Yu and Kumbier, 2019].