THE UNIVERSITY OF SYDNEY

Advanced Machine Learning

(COMP 5328)

Week 7 Tutorial:
Sparse Coding and Regularisation

Anjin Liu
anjin.liu@sydney.edu.au

https://github.com/Anjin-Liu/usyd_comp_5328_tutorial_S22025

Tutorial Contents

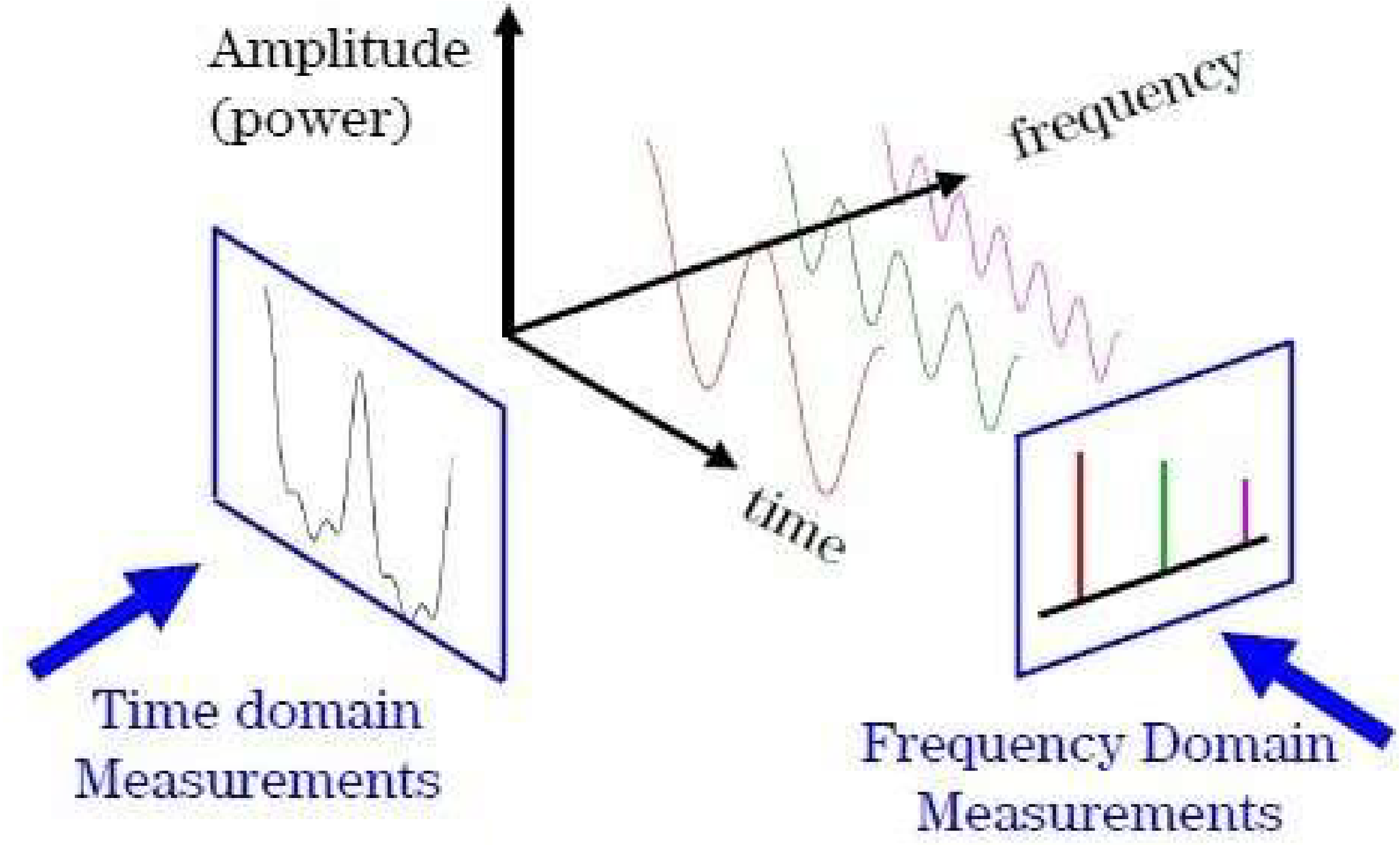
THE UNIVERSITY OF SYDNEY

- Review (20min):
  - Lecture 5: Sparse Coding and Regularisation
- Tutorial exercise & QA (40min):

Key points

- Sparse Coding
- Regularisation and Algorithmic Stability

Why Sparse?



Observations

Fundamental Structure

Why Sparse?

THE UNIVERSITY OF SYDNEY

We want sparse representations to uncover the **fundamental structure** of the data, rather than merely **fitting all observed variations with a complex model**.

Sparse=simpler, more interpretable representation

Why Sparse?

THE UNIVERSITY OF SYDNEY

Efficient representation
Noise robustness
Learning meaningful features
Interpretability
…

Why Sparse?

Time	Method / Model	Core Idea / Contribution	Relation to Sparsity
1990s	**Sparse Coding / Dictionary Learning**	Learn a dictionary D so each sample is a sparse linear combination of dictionary atoms	**Explicit sparsity**: each sample uses only a few basis elements
1997	**L1 Regularization / Lasso**	Achieve sparsity via L1 penalty	L1 constraint encourages sparse coefficients; idea inspired sparse coding
2006	**Deep Belief Networks (DBN)**	Unsupervised multi-layer feature learning	Introduced **sparse activations** in hidden layers
2010	**Convolutional Sparse Coding / Early CNNs**	Combine convolution with sparse coding for images	**Local sparse features**, similar to CNN filters
2012	**AlexNet (CNN)**	Deep convolutional network breakthrough on ImageNet	ReLU → **implicit sparsity**; Dropout → sparse training
2014	**VAE / Autoencoder**	Learn low-dimensional sparse representations	Sparse constraints on hidden layers → sparse feature encoding
2017	**Transformer (Attention)**	Use self-attention to extract sequence/image features	Sparse attention → only select important tokens/positions
2020+	**Vision Transformer / Sparse Transformers**	Large-scale Transformers for CV/NLP	**Explicit/implicit sparsity** throughout: sparse activations, sparse attention, sparse MLPs

Dictionary learning

Note that

$$\arg \min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2,$$

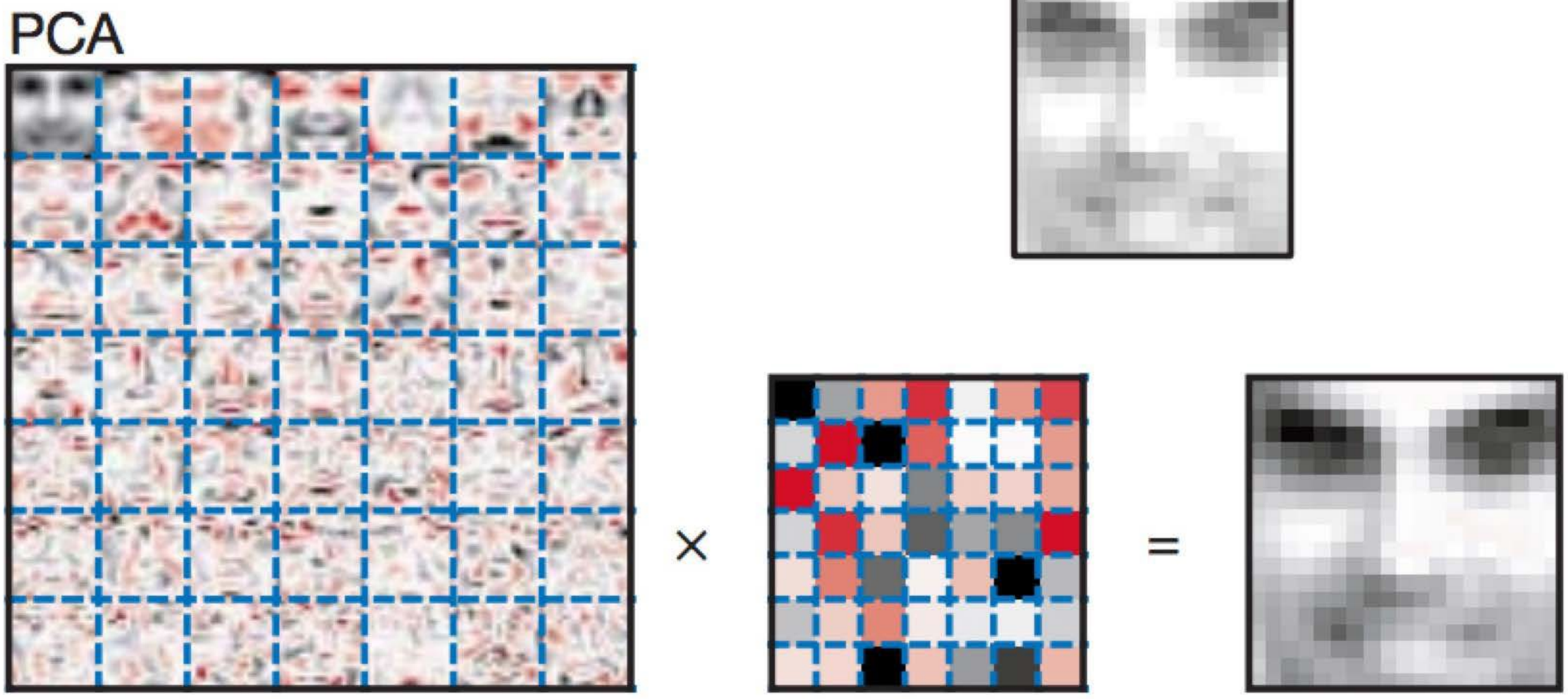
where $\mathcal{D}$ and $\mathcal{R}$ are some specific domains for D and R.

e.g., for PCA, orthogonal
for k–measn, one–hot s.t. one centroid per data
for NMF, non–negative

Dictionary learning

PCA: $A = U\Lambda U^T$

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^k} \|x - D\alpha\|^2.$$

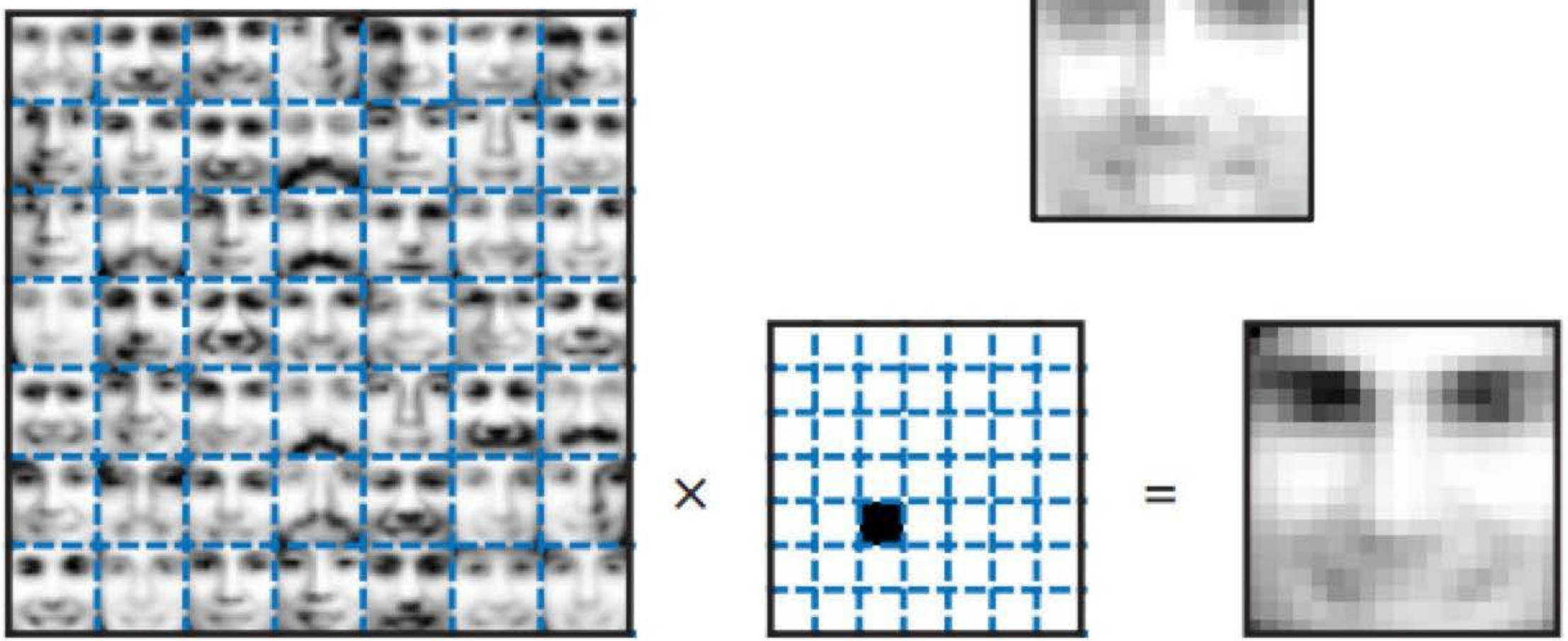


Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." Nature 401.6755 (1999): 788.

Dictionary learning

K-means clustering:

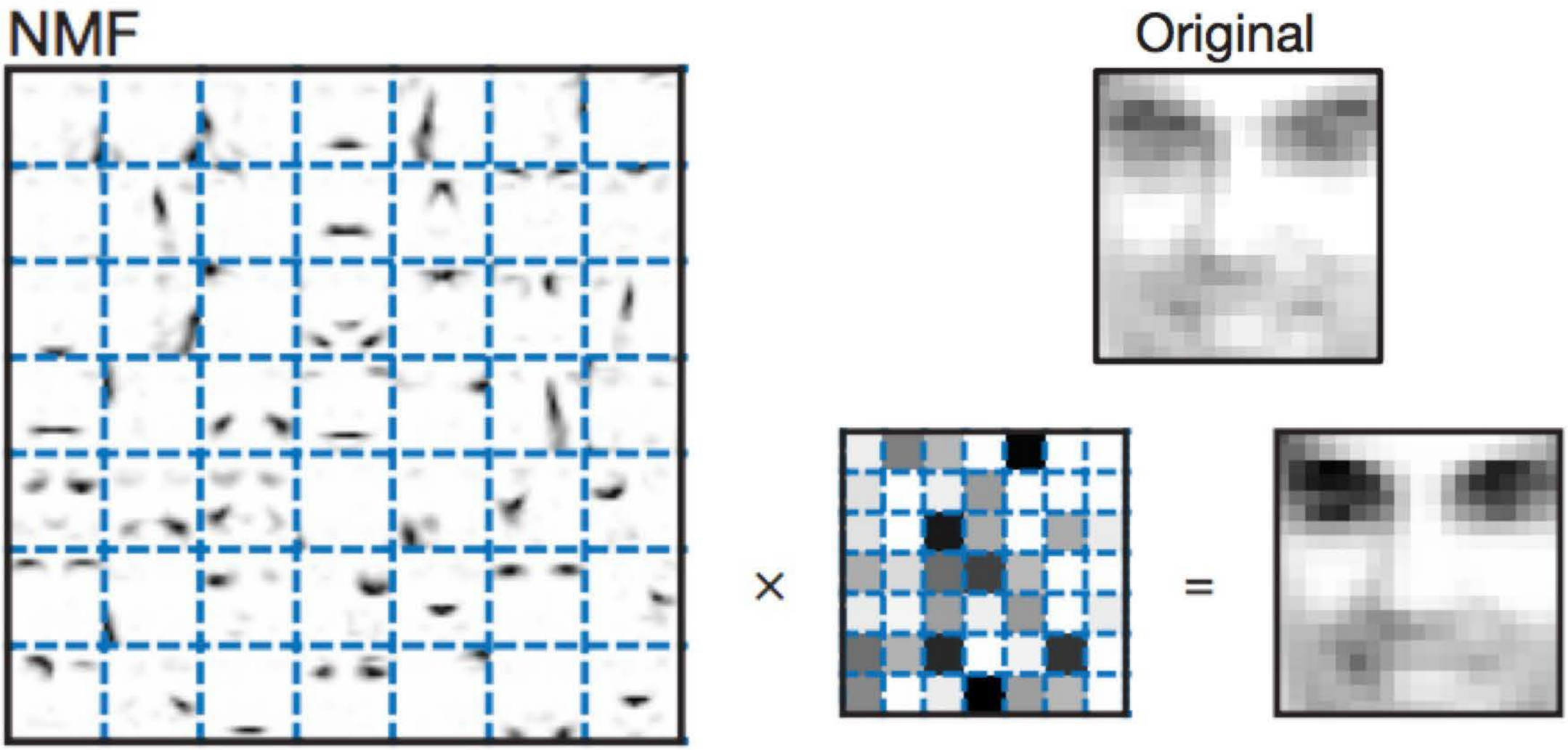
$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^k} \|x - D\alpha\|^2.$$



Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." Nature 401.6755 (1999): 788.

Non-negative matrix factorisation

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^k} \|x - D\alpha\|^2.$$

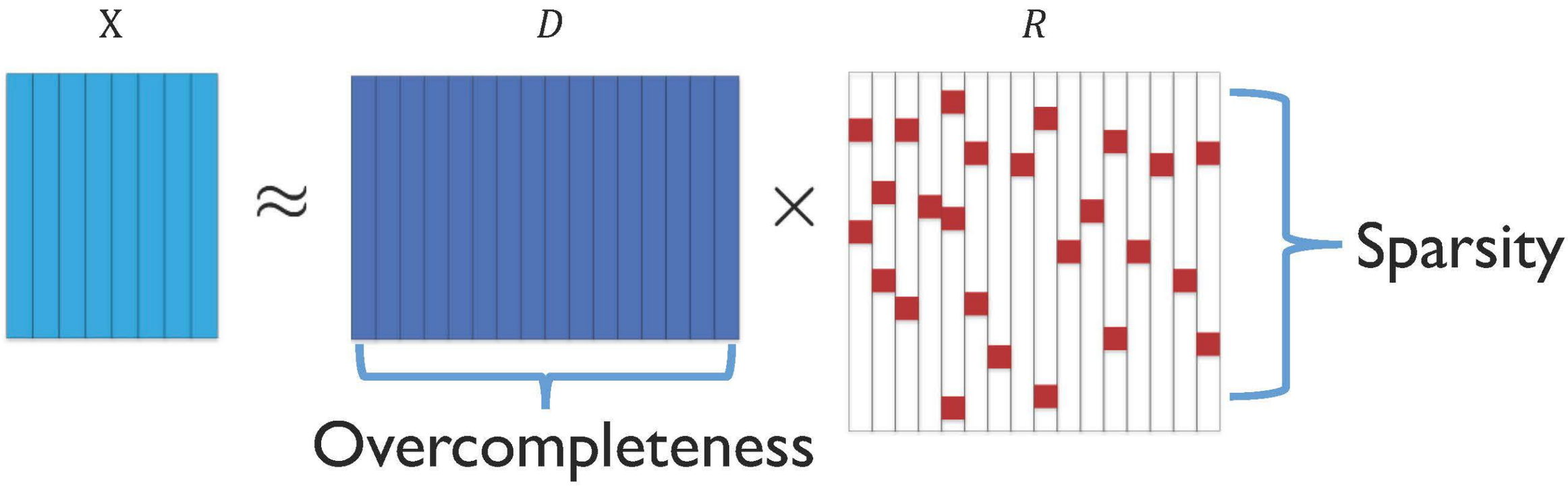


Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." Nature 401.6755 (1999): 788.

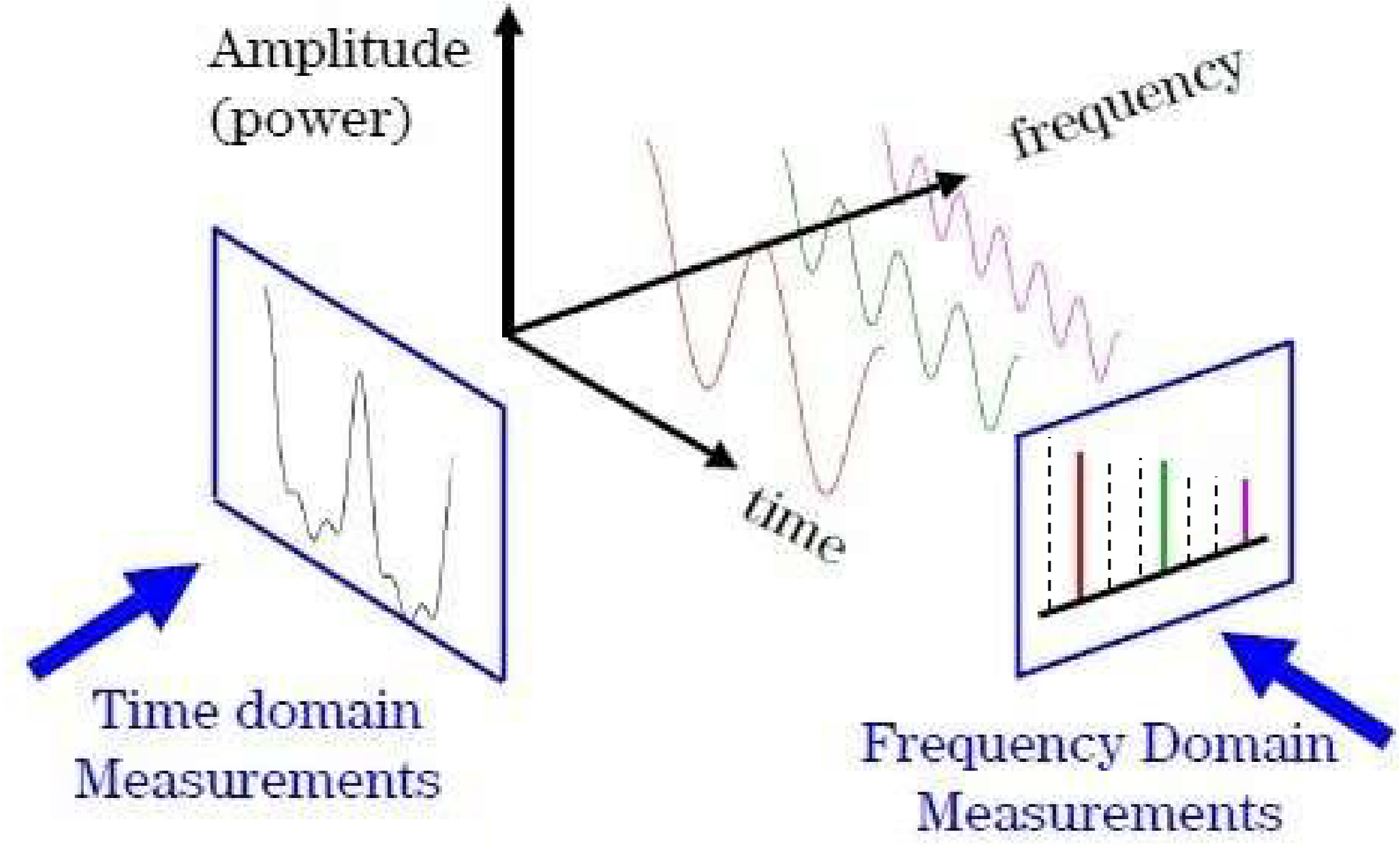
Sparse coding

Note that

$$\arg \min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2$$



Why Sparse?



Overcompleteness Dictionary

Measure of Sparsity

The objective function:

$$\min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2 + \lambda \psi(R)$$

Data fitting — Sparse regularisation

Question: how can we design the regularisation to make *R* to be sparse?

ℓ_p norm

ℓ_p norm: $\|\alpha\|_p = \left(\sum_{j=1}^{k} |\alpha_j|^p \right)^{1/p}$, where $\alpha \in \mathbb{R}^k$.

In other words, $\|\alpha\|_p^p = \sum_{j=1}^{k} |\alpha_j|^p$.

l_0 norm example

$x = [3,0,0,2,5]$

$3^0 + 0^0 + 0^0 + 2^0 + 5^0$

$1 + 0 + 0 + 1 + 1 = 3$

p=0: how many entries are non-zero

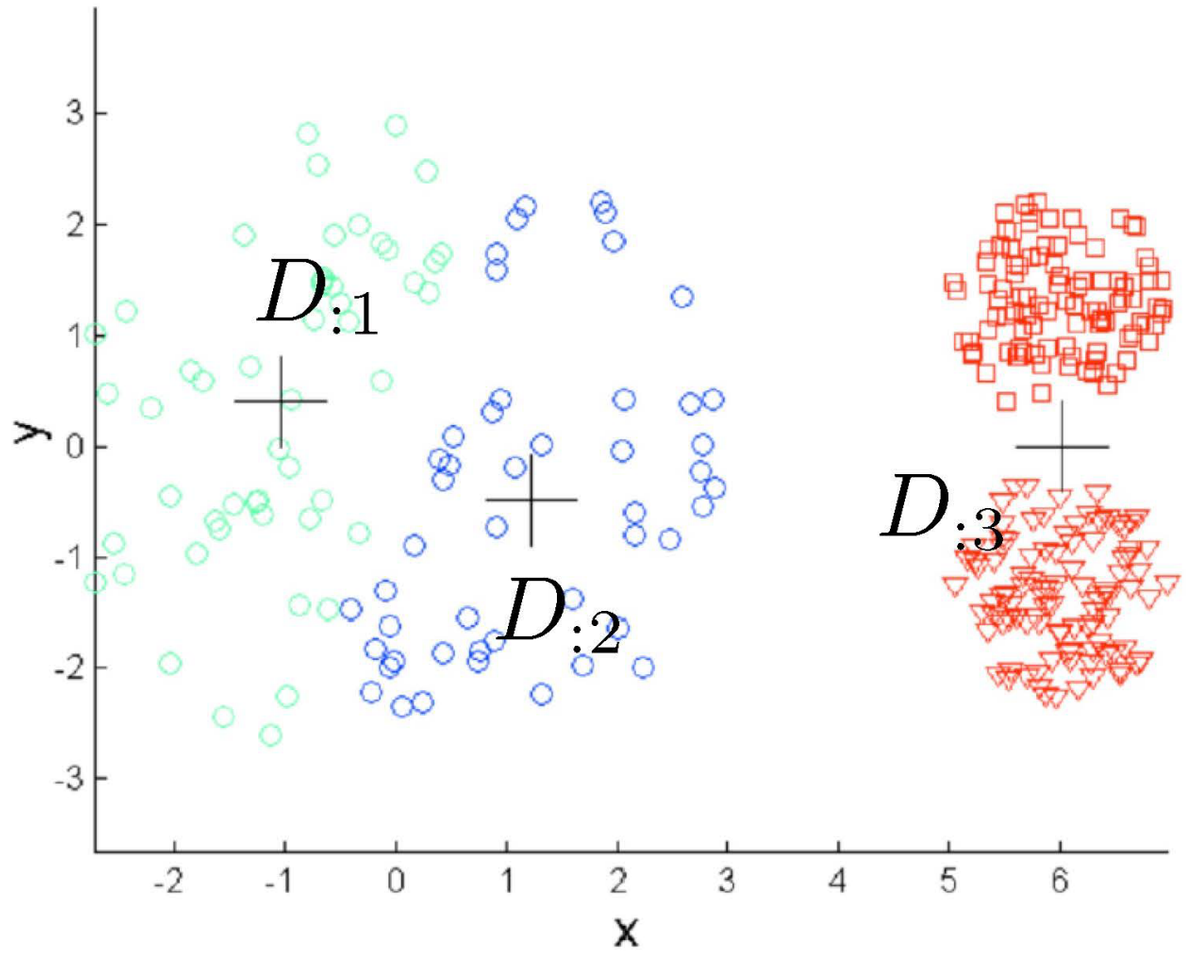
p=1, absolute value

p=2, sum of squares

K-means

K-means clustering:

$$\min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2$$



Special requirement: each column of R is an one-hot vector, i.e., $\|R_i\|_0 = 1 \ \& \ \|R_i\|_1 = 1.$

K-SVD

K-SVD: $$\min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|^2$$

Special requirement: each column of R is a sparse vector, i.e., $\|R_i\|_0 \leq k' << k$.

number much smaller than k (the dimension of R)

Measure of Sparsity

The objective function:

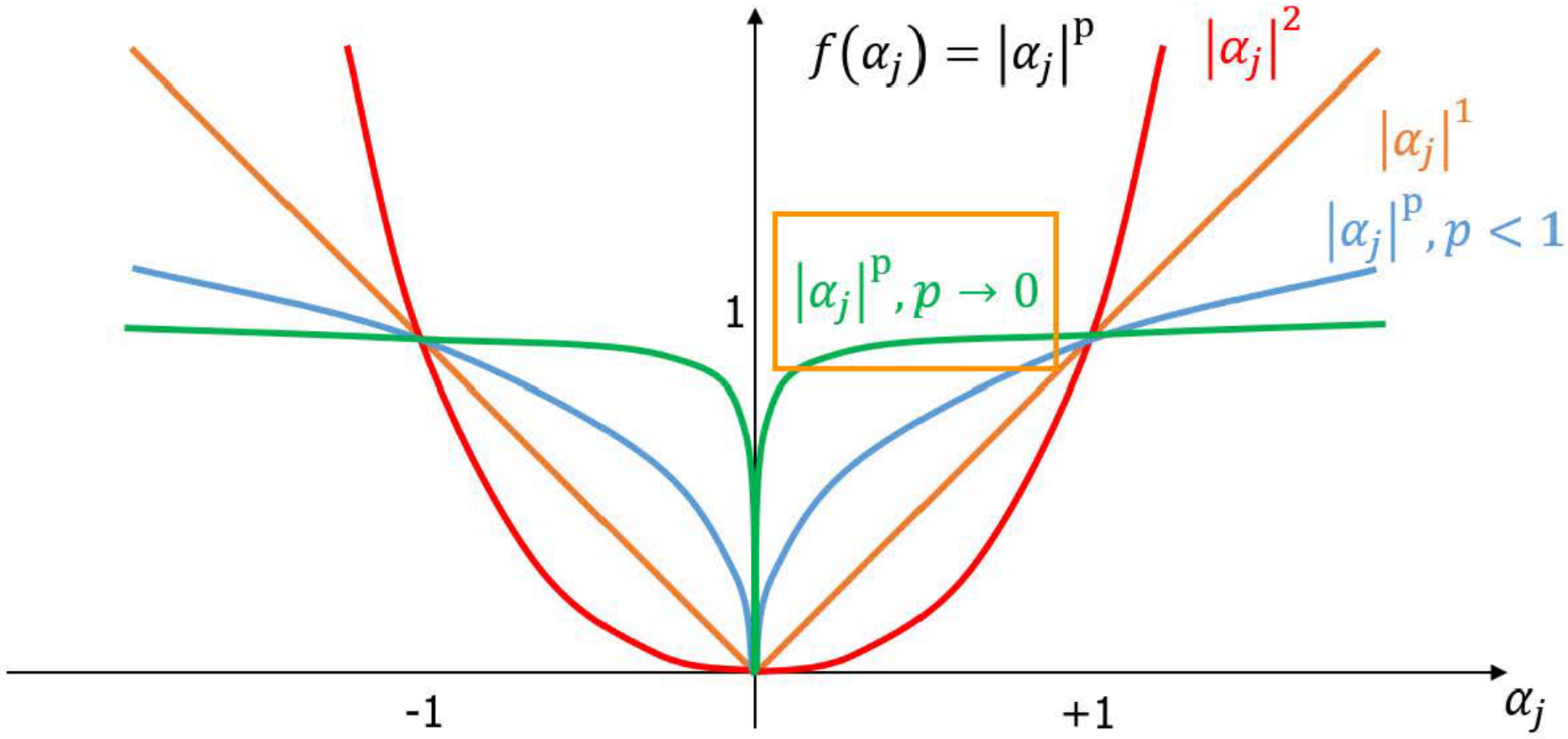
$$\min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2 + \lambda \psi(R)$$

Data fitting — Sparse regularisation

Question: how can we design the regularisation to make *R* to be sparse?

Measure of Sparsity: ℓ_0 norm

$$\|\alpha\|_p^p = \sum_{j=1}^{k} |\alpha_j|^p.$$



As $p \to 0$, we get a count of the non-zeros in the vector.

So we can employ $\|\alpha\|_0$ to measure sparsity.

Measure of Sparsity: ℓ_0 norm

As $p \to 0$, we get a count of the non-zeros in the vector.
So we can employ $\|\alpha\|_0$ to measure sparsity.

However, the ℓ_0 minimisation is not easy. How to do?

the gradient is not easy to calculated

Sparse coding learning algorithms

The ℓ_0 norm based approaches:

- $\min_{\alpha} \|X - D\alpha\|_F^2 \ \ s.t. \forall\, i, \|\alpha\|_0 < L.$
- $\min_{\alpha} \|\alpha\|_0 \ \ \text{s.t.}\ \|X - D\alpha\|_F^2 \leq \epsilon.$

there are also some non-gradient-based optimization methods

Greedy algorithms:

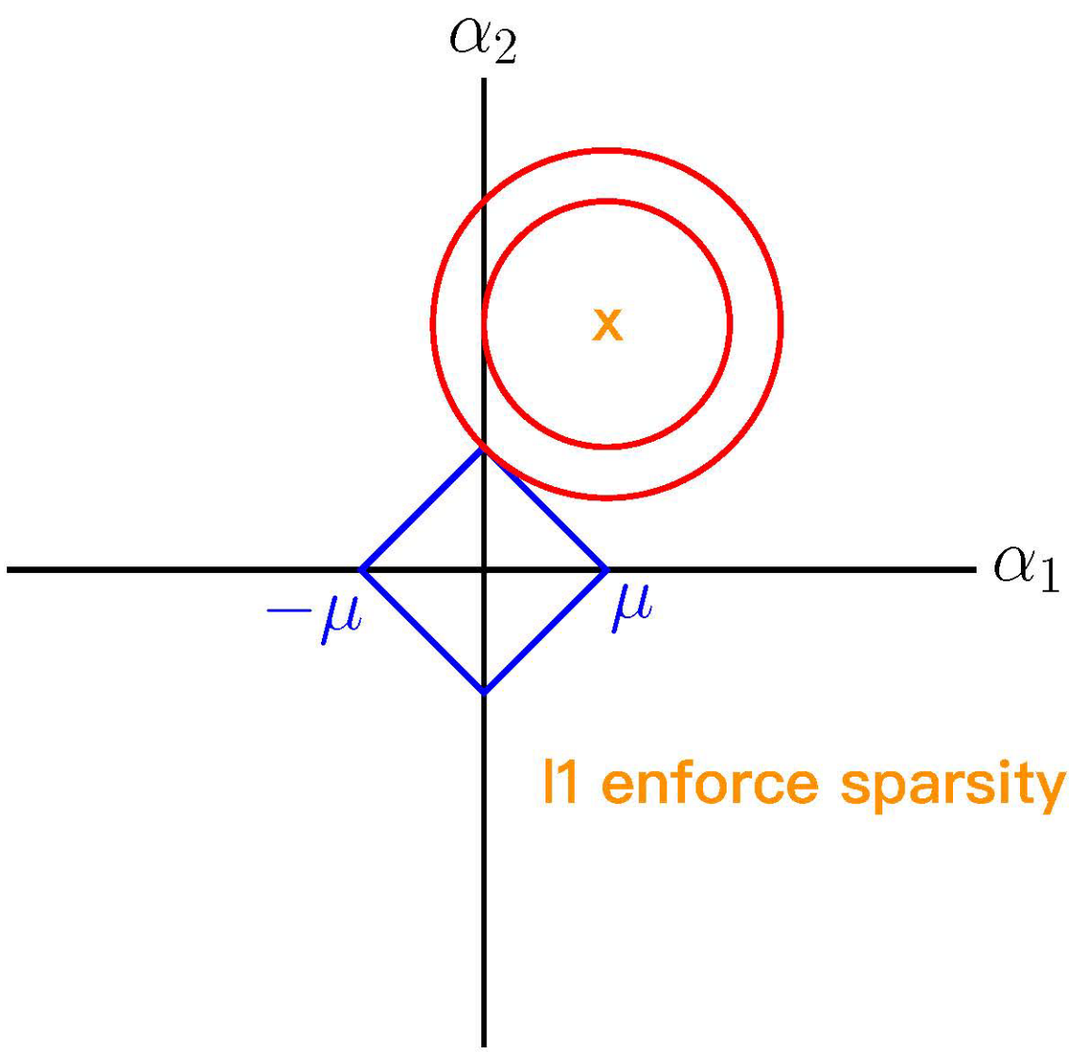
- OMP (Y. Pati, et al. 1993; J. Tropp 2004).
- Subspace pursuit (SP) (W. Dai and O. Milenkovic 2009), CosaMP (D. Needell and J. Tropp 2009).
- IHT (T. Blumensath and M. Davies 2009).

Measure of Sparsity ℓ_1 norm

Besides, we can use L1

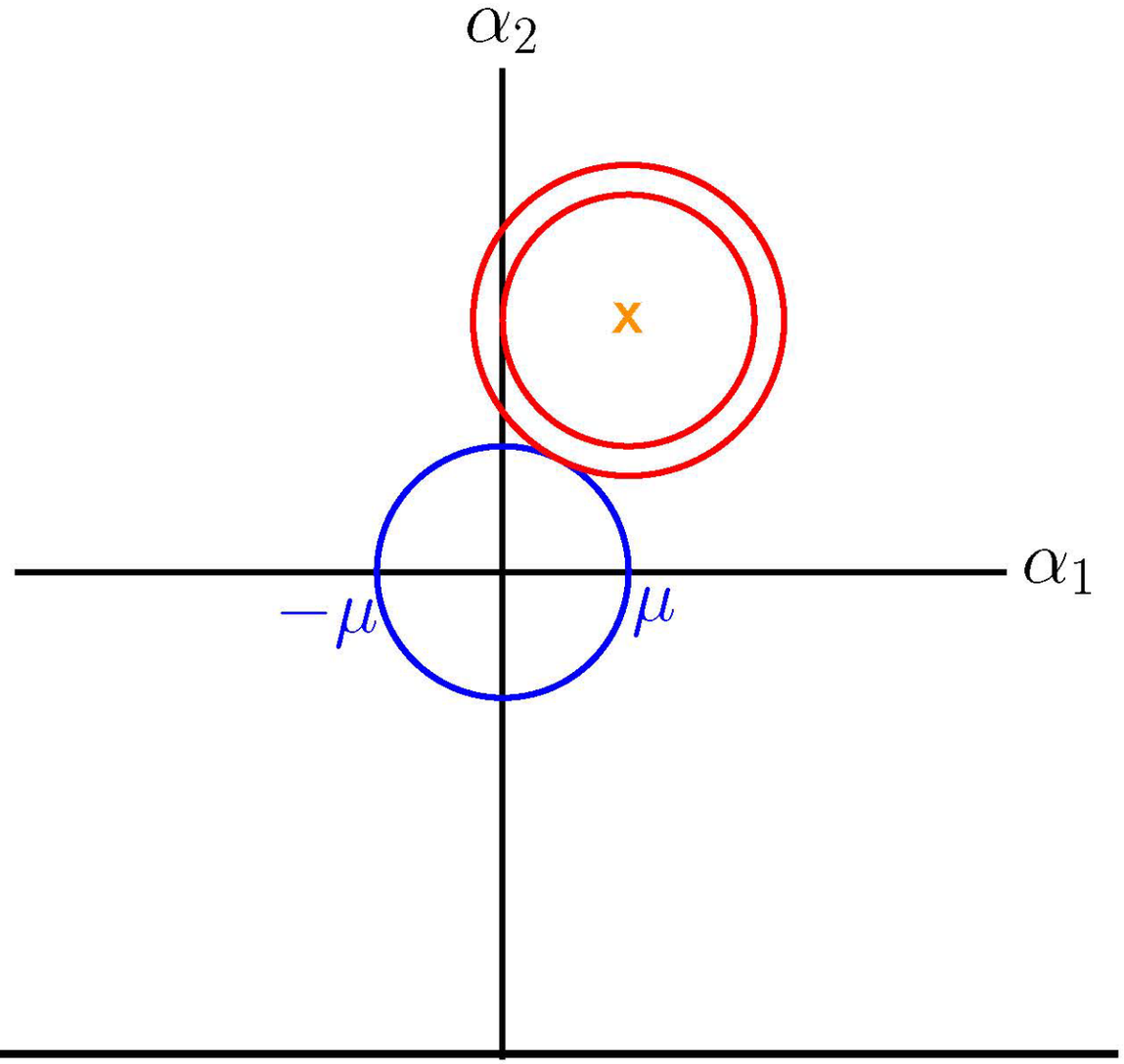
2D example (compared with ℓ_2-norm):

$$\min_{\alpha} \frac{1}{2}\|x-\alpha\|_2^2 \;\; s.t. \|\alpha\|_1 \leq \mu$$



l1 enforce sparsity

$$\min_{\alpha} \frac{1}{2}\|x-\alpha\|_2^2 \;\; s.t. \|\alpha\|_2 \leq \mu$$



Sparse coding learning algorithms

The ℓ_1 norm based approaches:

hard constraint

- $\min_{\alpha} \|\alpha\|_1 \ \ \text{s.t.} \ \|X - D\alpha\|_F^2 \leq \epsilon.$

- $\min \|X - D\alpha\|_F^2 + \lambda \|\alpha\|_1.$ soft constraint

Bayesian approach:

- Relevance vector machine (RVM) (M. Tipping 2001).
- Bayesian compressed sensing (BCS) (S. Ji, et al. 2008).

THE UNIVERSITY OF SYDNEY

Regularisation and algorithmic stability

slightly chance the input, the output will not change significantly

No-Free-Lunch Theorem

THE UNIVERSITY OF SYDNEY

- Sparse algorithms are not stable!
- A learning algorithm is said to be stable if slight perturbations in the training data result in small changes in the output of the algorithm, and these changes vanish as the data set grows bigger and bigger.

Xu, H., Caramanis, C., & Mannor, S. (2012). Sparse algorithms are not stable: A no-free-lunch theorem. IEEE transactions on pattern analysis and machine intelligence, 34(1), 187-193.

Algorithmic Stability

THE UNIVERSITY OF SYDNEY

We have two different training samples:

$$S = \{(X_1, Y_1), \ldots, (X_{i-1}, Y_{i-1}), (X_i, Y_i), (X_{i+1}, Y_{i+1}), \ldots, (X_n, Y_n)\}$$

$$S^i = \{(X_1, Y_1), \ldots, (X_{i-1}, Y_{i-1}), (X_i', Y_i'), (X_{i+1}, Y_{i+1}), \ldots, (X_n, Y_n)\}$$

They are different because of only one training example.

An algorithm is uniformly stable if for any example (X, Y)

$$|\ell(X, Y, h_S) - \ell(X, Y, h_{S^i})| \leq \epsilon(n).$$

Note that $\epsilon(n)$ will vanish as n goes to infinity.

Generalisation error

$$
\begin{aligned}
R(h_S) - \min_{h \in H} R(h) &= R(h_S) - R(h^*) \\
&= R(h_S) - R_S(h_S) + R_S(h_S) - R_S(h^*) + R_S(h^*) - R(h^*) \\
&\leq R(h_S) - R_S(h_S) + R_S(h^*) - R(h^*) \\
&\leq |R(h_S) - R_S(h_S)| + |R(h^*) - R_S(h^*)| \\
&\leq \sup_{h \in H} |R(h) - R_S(h)| + \sup_{h \in H} |R(h) - R_S(h)| \\
&= 2 \sup_{h \in H} |R(h) - R_S(h)|.
\end{aligned}
$$

$$
R(h_S) - R_S(h_S) \leq \sup_{h \in H} |R(h) - R_S(h)|.
$$

Algorithmic Stability

Proof not examined

THE UNIVERSITY OF SYDNEY

A good stable algorithm will have a good generalisation ability:

$$\mathbb{E}[R(h_S) - R_S(h_S)]$$

$$= \mathbb{E}_S \left[\mathbb{E}_{X,Y}[\ell(X, Y, h_S)] - \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h_S) \right]$$

$$= \mathbb{E}_S \left[\mathbb{E}_{S'} \left[\frac{1}{n} \sum_{i=1}^{n} \ell(X'_i, Y'_i, h_S) \right] - \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h_S) \right]$$

$$= \mathbb{E}_{S,S'} \left[\frac{1}{n} \sum_{i=1}^{n} (\ell(X'_i, Y'_i, h_S) - \ell(X_i, Y_i, h_S)) \right]$$

$$= \mathbb{E}_{S,S'} \left[\frac{1}{n} \sum_{i=1}^{n} (\ell(X'_i, Y'_i, h_S) - \ell(X'_i, Y'_i, h_{S^i})) \right]$$

$$\leq \epsilon'(n)$$

if not stable, it doesn't means cannot generalize well. (I1 can make it generalize well

$\epsilon'(n)$ would be small if the learning algorithm is stable (because h_S and h_{S^i} are similar). This implies that stable algorithms will have small expected generalisation errors.

where $S' = \{(X'_1, Y'_1), \ldots, (X'_n, Y'_n)\}$.

Regularisation and Stability

THE UNIVERSITY OF SYDNEY

ℓ_2 norm regularisation will make learning algorithms stable if the employed surrogate loss function is convex.

Proof not examined

$$h_S = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h) + \lambda \|h\|_2^2$$

If the convex surrogate loss function is L-Lipschitz continuous w.r.t. h, $\|X\|_2 \leq B$, we have

$$|\ell(X, Y, h_S) - \ell(X, Y, h_{S^i})| \leq \frac{2L^2B^2}{\lambda n}.$$

Key points

- L-p norm
- K-means, K-SVD
- Measure of Sparsity (p=0, p=1)
- Stability