Machine Learning and Numerical Analysis

Francis Bach

Willow project, INRIA - Ecole Normale Supérieure

November 2010

Machine Learning and Numerical Analysis
Outline

- **Machine learning**
  - Supervised vs. unsupervised
- **Convex optimization for supervised learning**
  - Sequence of linear systems
- **Spectral methods for unsupervised learning**
  - Sequence of singular value decompositions
- **Combinatorial optimization**
  - Polynomial-time algorithms and convex relaxations

Statistical machine learning
Computer science and applied mathematics

- **Modelisation, prediction and control from training examples**
- **Theory**
  - Analysis of statistical performance
- **Algorithms**
  - Numerical efficiency and stability
- **Applications**
  - Computer vision, bioinformatics, neuro-imaging, text, audio

Statistical machine learning - Supervised learning

- Data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Goal**: predict $y \in \mathcal{Y}$ from $x \in \mathcal{X}$, i.e., find $f : \mathcal{X} \to \mathcal{Y}$
- Empirical risk minimization

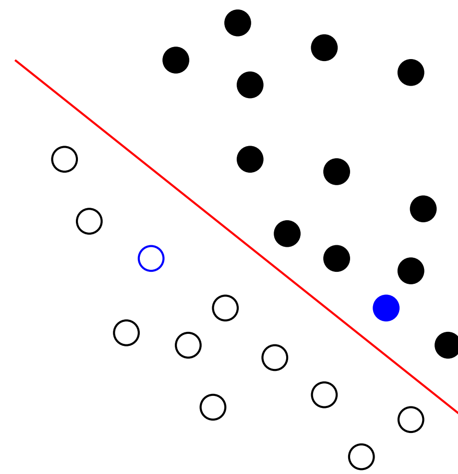
$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)) \quad + \quad \frac{\lambda}{2}\|f\|^2$$

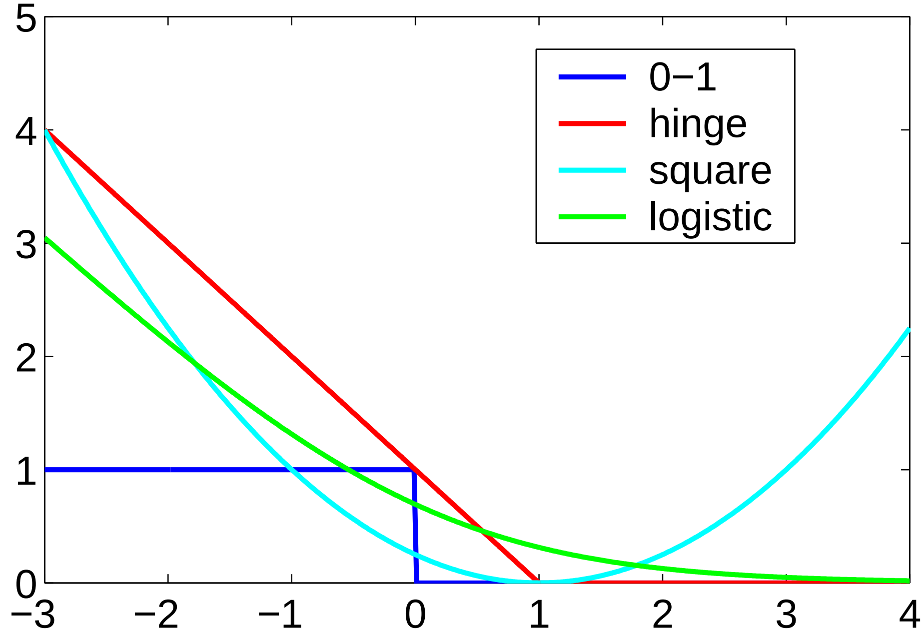
Data-fitting + Regularization

- **Scientific objectives**:
  - Studying generalization error
  - Improving calibration
  - Choosing appropriate representations - selection of appropriate loss
  - Two main types of norms: ℓ_2 **vs.** ℓ_1

Usual losses

- **Regression**: $y \in \mathbb{R}$, prediction $\hat{y} = f(x)$,
  - quadratic cost $\ell(y, f(x)) = \frac{1}{2}(y - f(x))^2$
- **Classification** : $y \in \{-1, 1\}$ prediction $\hat{y} = \text{sign}(f(x))$
  - loss of the form $\ell(y, f(x)) = \ell(yf(x))$
  - "True" cost: $\ell(yf(x)) = 1_{yf(x)<0}$
  - Usual convex costs:





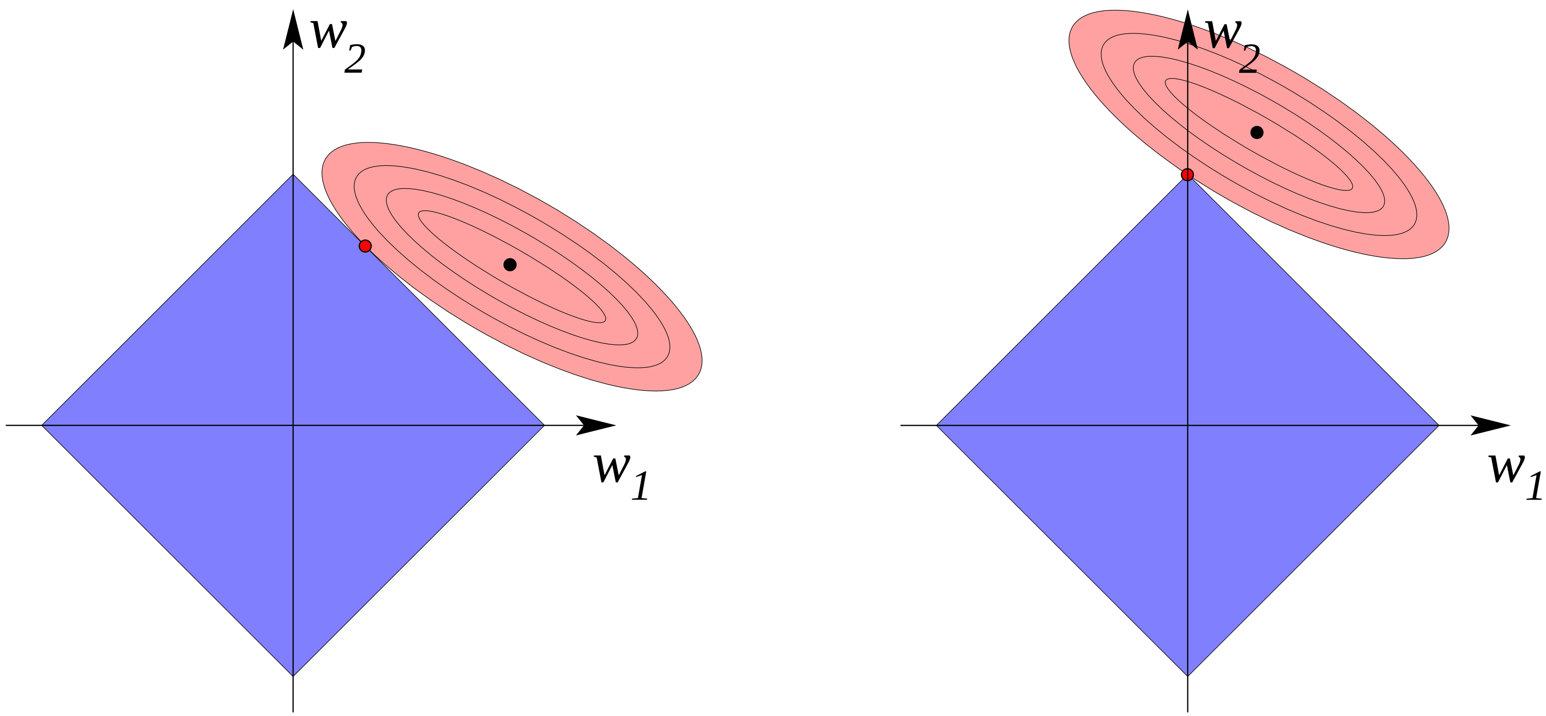
Supervised learning - Parsimony and ℓ_1-norm

- Data $(x_i, y_i) \in \mathbb{R}^p \times \mathcal{Y}$, $i = 1, \dots, n$

$$\min_{w \in \mathbb{R}^p} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) \quad + \quad \lambda \sum_{j=1}^{p} |w_j|$$

Data-fitting + Regularization

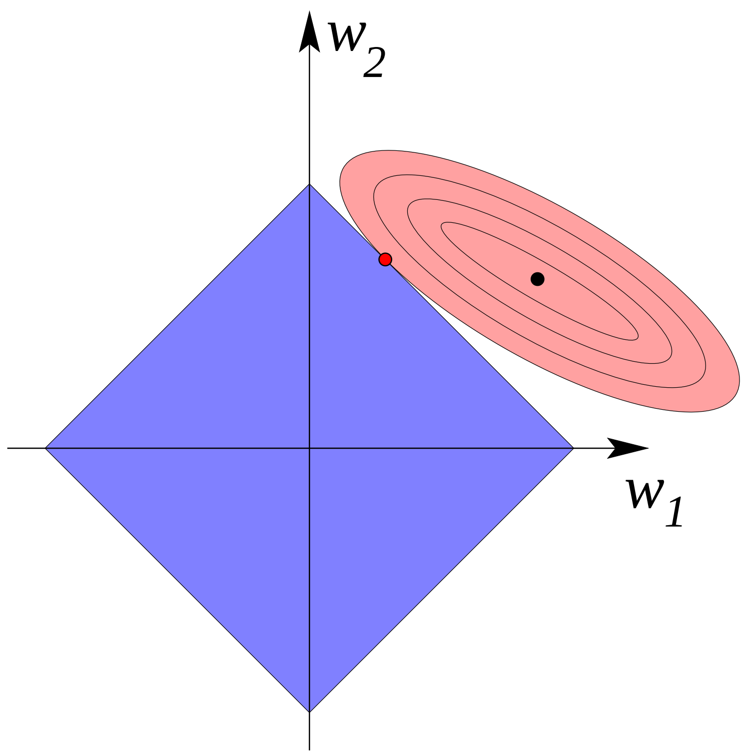
- At the optimum, w is in general **sparse**

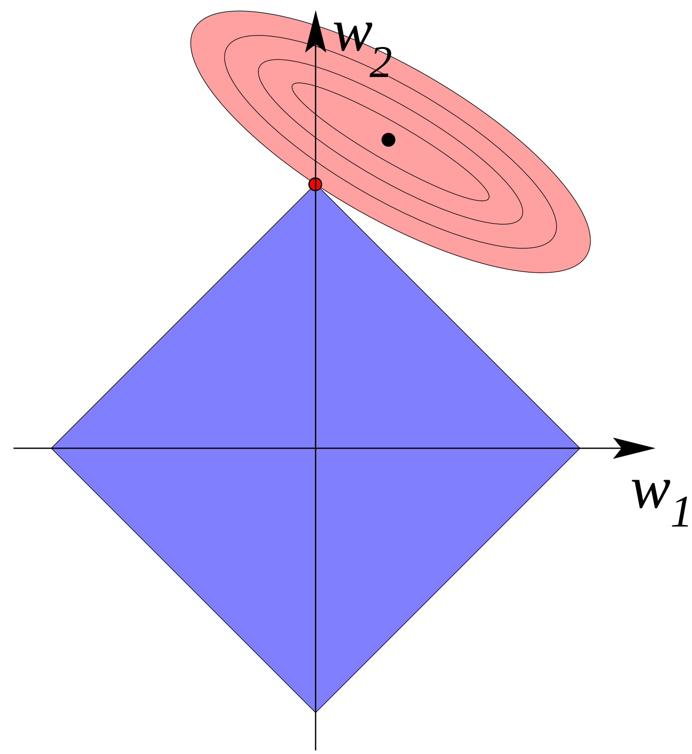


Sparsity in machine learning

- **Assumption**: $\mathbf{y} = \mathbf{w}^\top \mathbf{x} + \varepsilon$, with $w \in \mathbb{R}^p$ sparse
  - Proxy for interpretability
  - Allow high-dimensional inference: $\boxed{\log p = O(n)}$

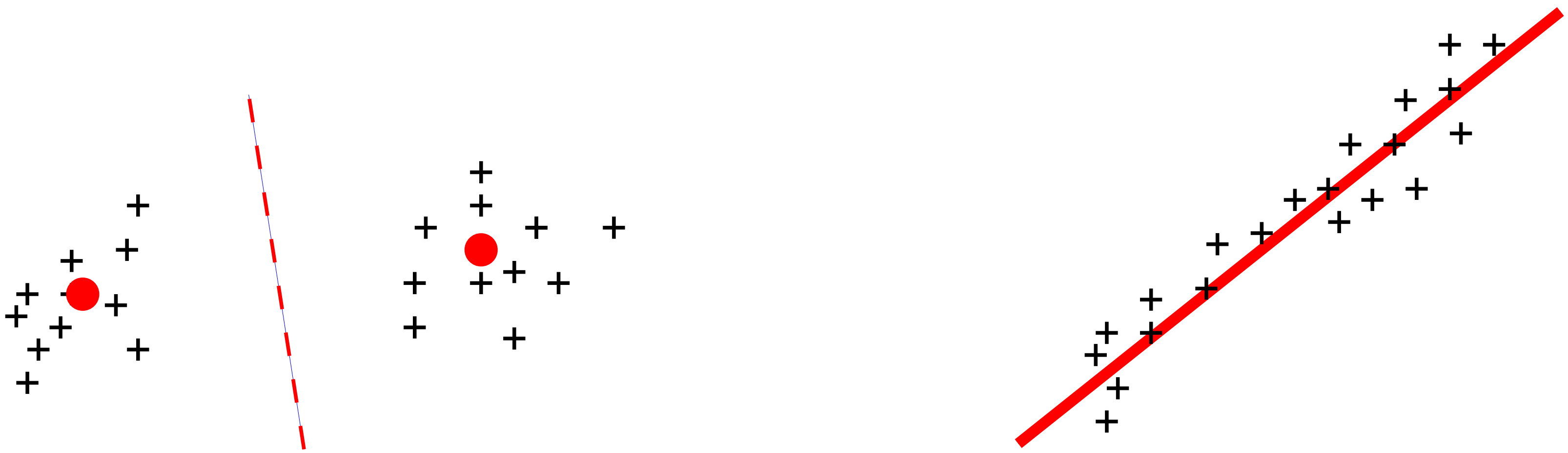
- **Sparsity and convexity** (ℓ_1-norm regularization): $\boxed{\min_{\mathbf{w} \in \mathbb{R}^p} L(\mathbf{w}) + \|\mathbf{w}\|_1}$





Statistical machine learning - Unsupervised learning

- Data $x_i \in \mathcal{X}$, $i = 1, \ldots, n$. **Goal: "Find" structure within data**
  - Discrete : clustering
  - Low-dimension : principal component analysis



Statistical machine learning - Unsupervised learning

- Data $x_i \in \mathcal{X}$, $i = 1, \ldots, n$. **Goal: "Find" structure within data**
  - Discrete : clustering
  - Low-dimension : principal component analysis

- **Matrix factorization**:

$$\boxed{X = DA}$$

  - Structure on D and/or A
  - Algorithmic and theoretical issues

- **Applications**

Learning on matrices - Collaborative filtering

- Given $n_{\mathcal{X}}$ "movies" $\mathbf{x} \in \mathcal{X}$ and $n_{\mathcal{Y}}$ "customers" $\mathbf{y} \in \mathcal{Y}$,
- predict the "rating" $z(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}$ of customer $\mathbf{y}$ for movie $\mathbf{x}$
- Training data: large $n_{\mathcal{X}} \times n_{\mathcal{Y}}$ incomplete matrix $\mathbf{Z}$ that describes the known ratings of some customers for some movies
- **Goal**: complete the matrix.

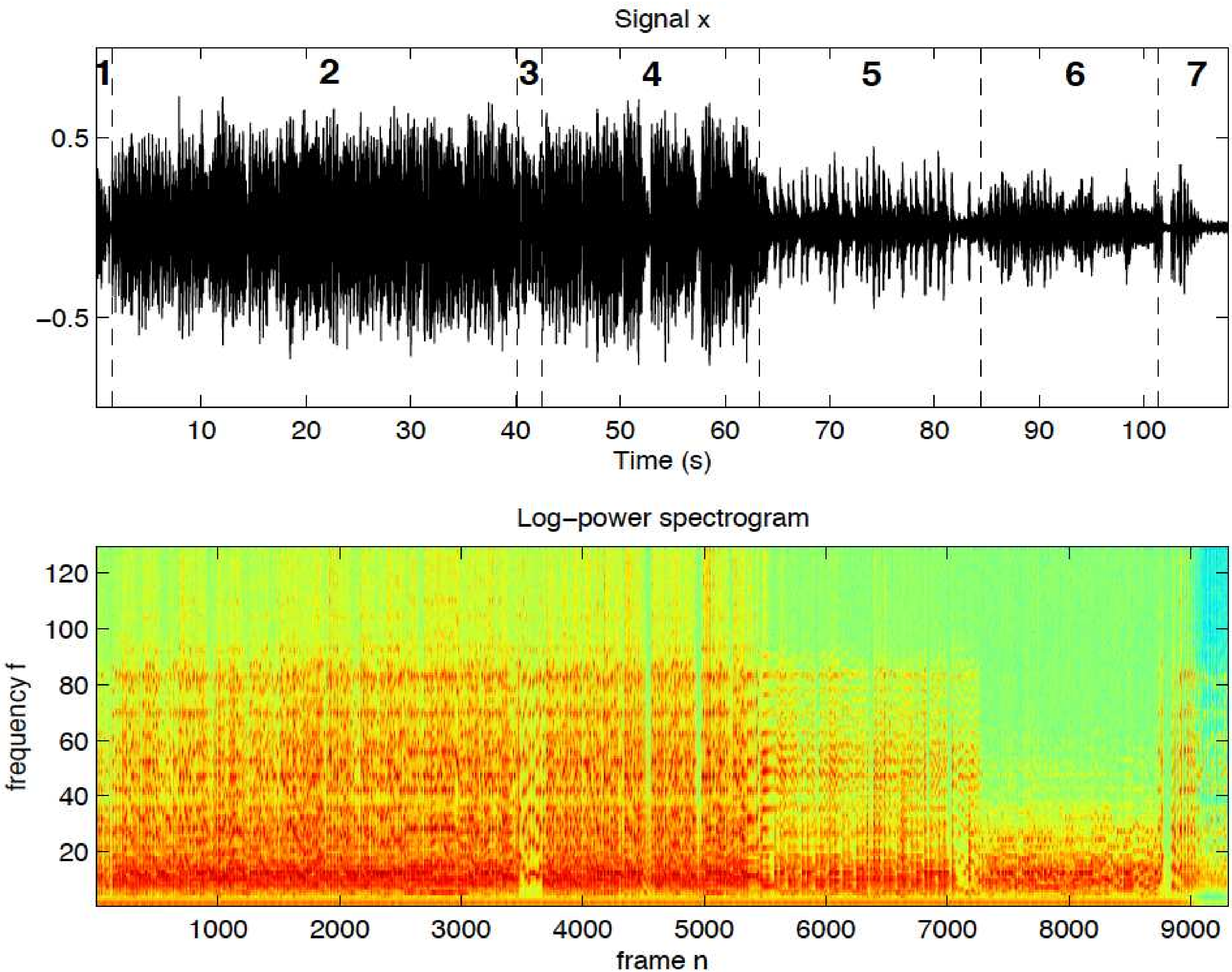
	1				1								2							
											2				1				3	
	2									3						3				3
						1			2					1				1		
		1		3							1						1	3		
	1		2													2				
								3												

Learning on matrices - Image denoising

- Simultaneously denoise all patches of a given image
- Example from Mairal et al. (2009)

Learning on matrices - Source separation

- Single microphone (Févotte et al., 2009)



Machine Learning and Numerical Analysis
Outline

- **Machine learning**
  - Supervised vs. unsupervised
- **Convex optimization for supervised learning**
  - Sequence of linear systems
- **Spectral methods for unsupervised learning**
  - Sequence of singular value decompositions
- **Combinatorial optimization**
  - Polynomial-time algorithms and convex relaxations

Supervised learning - Convex optimization

- Data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Goal**: predict $y \in \mathcal{Y}$ from $x \in \mathcal{X}$, i.e., find $f : \mathcal{X} \to \mathcal{Y}$
- Empirical risk minimization

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)) \quad + \quad \frac{\lambda}{2}\|f\|^2$$

Data-fitting $+$ Regularization

- **Typical problems**
  - f in vector space (e.g., $\mathbb{R}^p$)
  - ℓ convex with respect to second variable, potentially non smooth
  - Norm may be non differentiable
  - p and/or n large

Convex optimization - Kernel methods

- **Simplest case**: least-squares

$$\min_{w \in \mathbb{R}^p} \frac{1}{2n} \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

  – Solution: $w = (X^\top X + n\lambda I)^{-1} X^\top y$ in $O(p^3)$

- **Kernel methods**
  – Maybe re-written as $w = X^\top (XX^\top + n\lambda I)^{-1} y$ in $O(n^3)$
  – Replace $x_i^\top x_j$ by any positive definite *kernel function* $k(x_i, x_j)$, e.g., $k(x, x') = \exp(-\alpha \|x - x'\|_2^2)$

- **General losses** : Interior point vs. first order methods

- **Manipulation of large structured matrices**

Convex optimization - Low precision

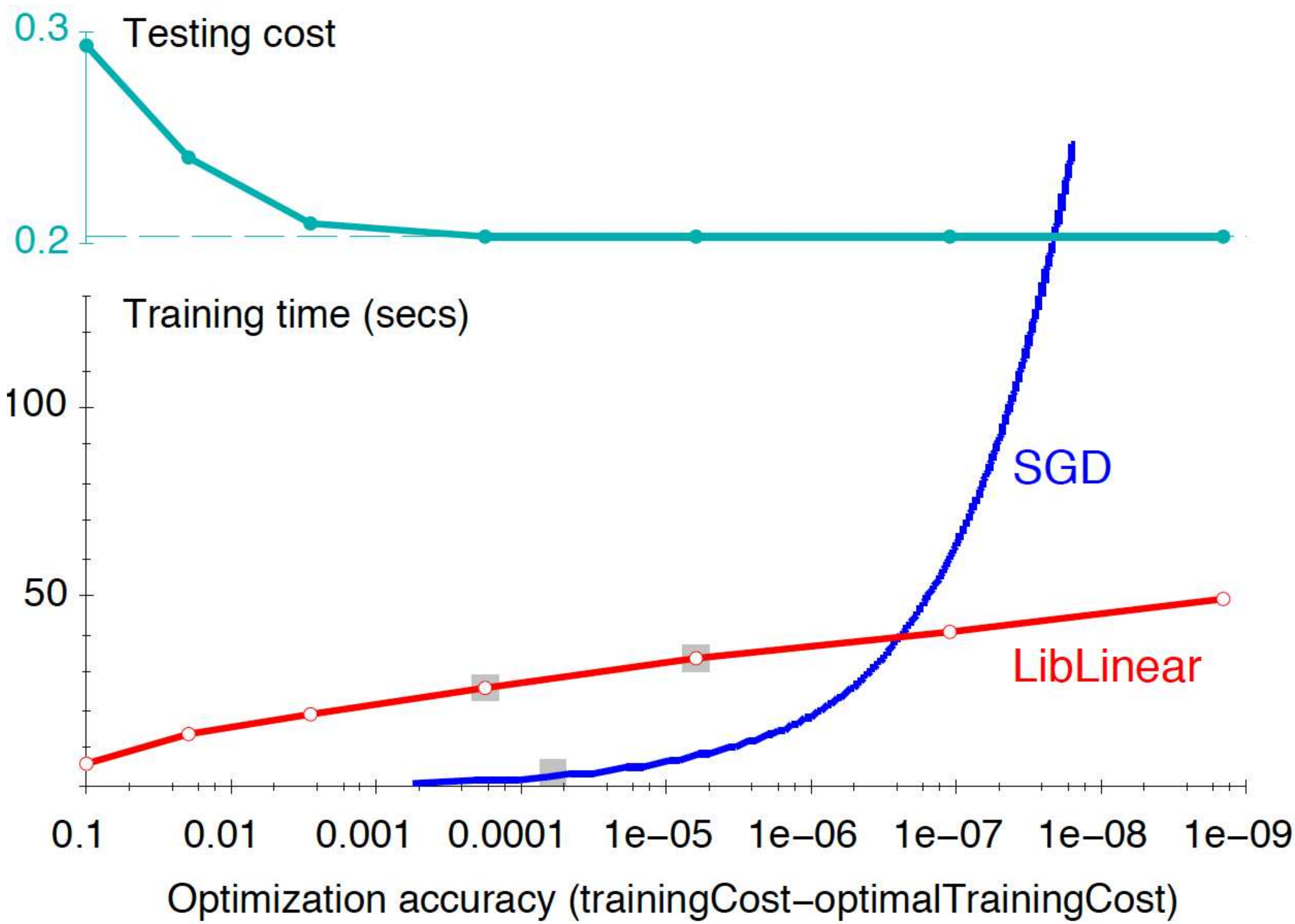
- Empirical risk minimization

$$\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i)) \quad + \quad \frac{\lambda}{2}\|f\|^2$$

Data-fitting + Regularization

- **No need to optimize below precision $n^{-1/2}$**
  - **Goal is to minimize test error**
  - Second-order methods adapted to high precision
  - First-order methods adapted to low precision

Convex optimization - Low precision (Bottou and Bousquet, 2008)



Convex optimization - Sequence of problems

- Empirical risk minimization

$$\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i)) \quad + \quad \frac{\lambda}{2}\|f\|^2$$

$$\text{Data-fitting} \quad + \quad \text{Regularization}$$

- **In practice**: Needs to be solved for many values of λ
- **Piecewise-linear paths**
  - In favorable situations
- **Warm restarts**

Convex optimization - First order methods

- Empirical risk minimization

$$\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i)) \quad + \quad \lambda\Omega(f)$$

$$\text{Data-fitting} \quad + \quad \text{Regularization}$$

- **Proximal methods** adapted to **non-smooth norms** and **smooth losses**
  - Need to solve **efficiently** problems of the form

$$\min_f \|f - f_0\|^2 + \lambda\Omega(f)$$

- **Stochastic gradient**: $\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i))$ proxy for $\mathbb{E}\ell(y, f(x))$

Machine Learning and Numerical Analysis
Outline

- **Machine learning**
  - Supervised vs. unsupervised
- **Convex optimization for supervised learning**
  - Sequence of linear systems
- **Spectral methods for unsupervised learning**
  - Sequence of singular value decompositions
- **Combinatorial optimization**
  - Polynomial-time algorithms and convex relaxations

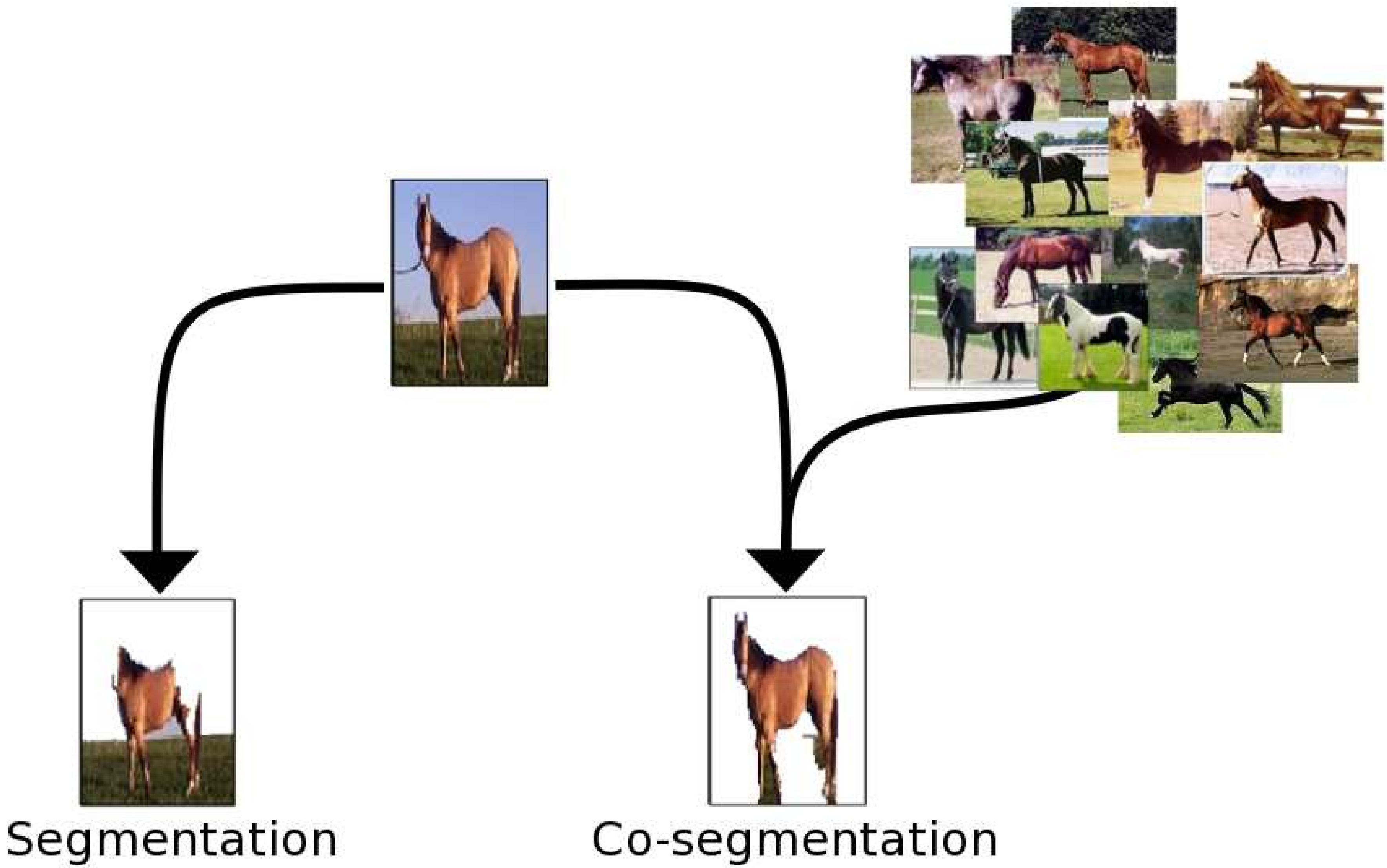
Unsupervised learning - Spectral methods

- **Spectral clustering**: given similarity matrix $W \in \mathbb{R}_+^{n \times n}$
  - Compute Laplacian matrix $L = \mathrm{Diag}(W1) - W = D - W$
  - Compute generalized eigenvector of (L, D)
  - May be seen as relaxation of normalized cuts

- **Applications**
  - Computer vision
  - Speech separation

Application to computer vision

Co-segmentation (Joulin et al., 2010)

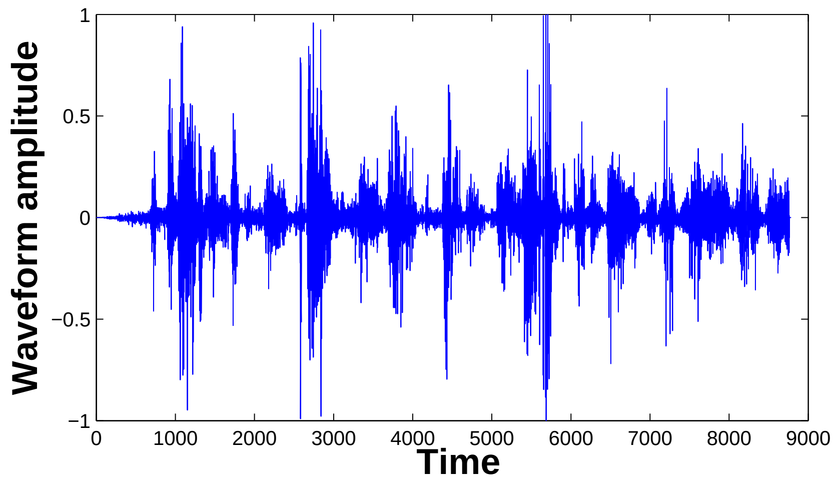


Blind one-microphone speech separation (Bach and Jordan, 2005)

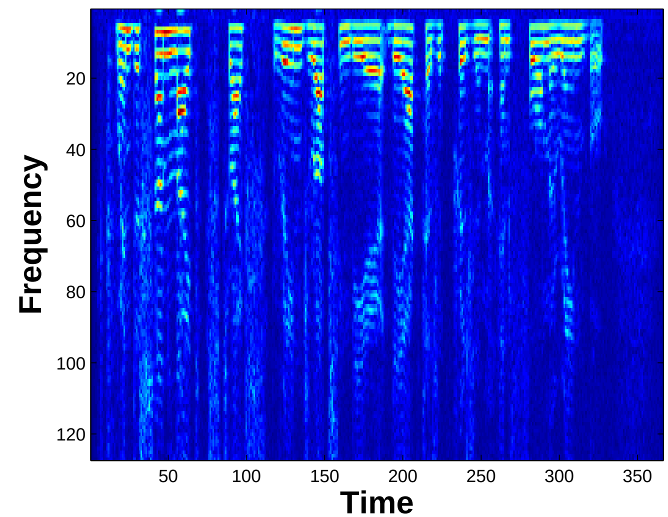
- Two or more speakers $s_1, \ldots, s_m$ - one microphone x
- Ideal acoustics $x = s_1 + s_2 + \cdots + s_m$
- **Goal**: recover $s_1, \ldots, s_m$ from x
- **Blind**: without knowing the speakers in advance
- **Formulation as spectogram segmentation**

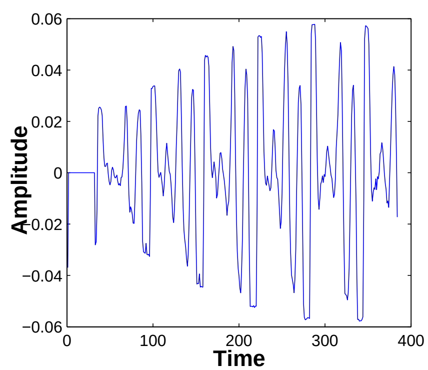
Spectrogram

- Spectrogram (a.k.a Gabor analysis, Windowed Fourier transforms)
  - cut the signals in overlapping frames
  - apply a window and compute the FFT



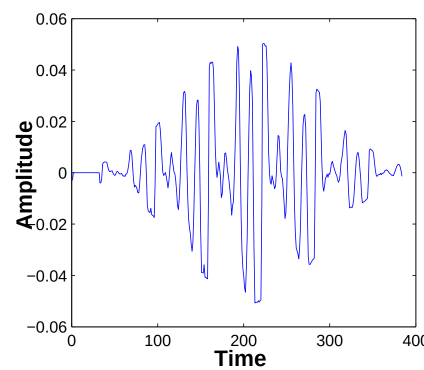
⇒





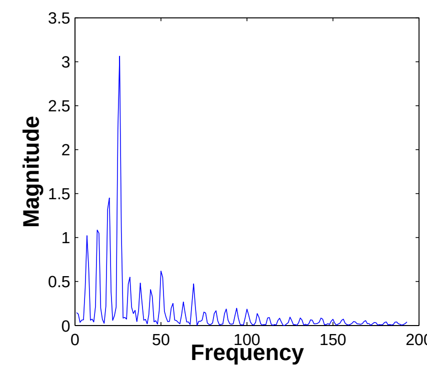
Windowing

⇒



Hamming window

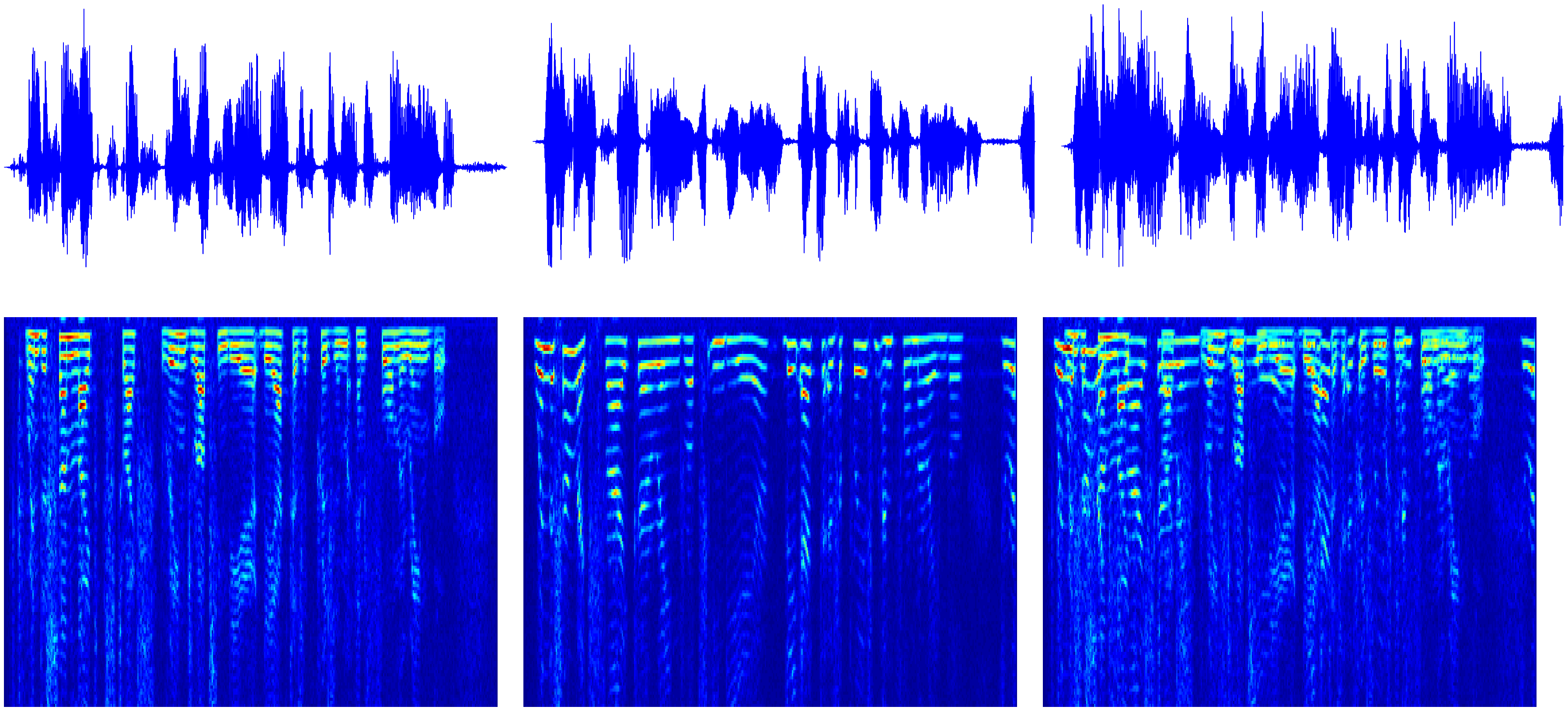
⇒



Fourier transform

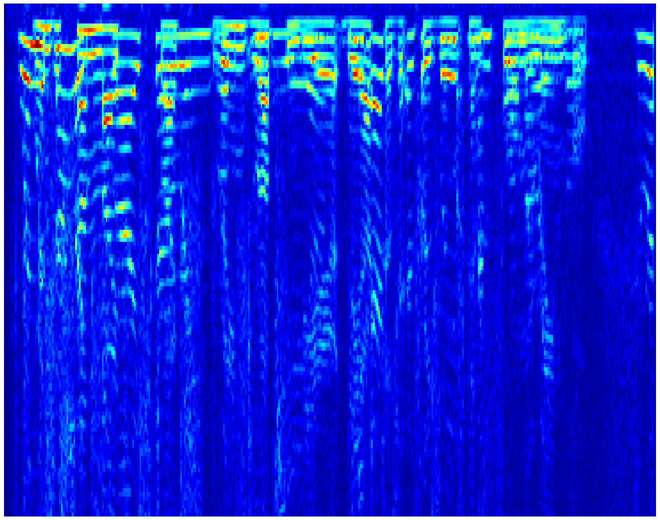
Sparsity and superposition

$$s_1 \quad + \quad s_2 \quad = \quad x$$

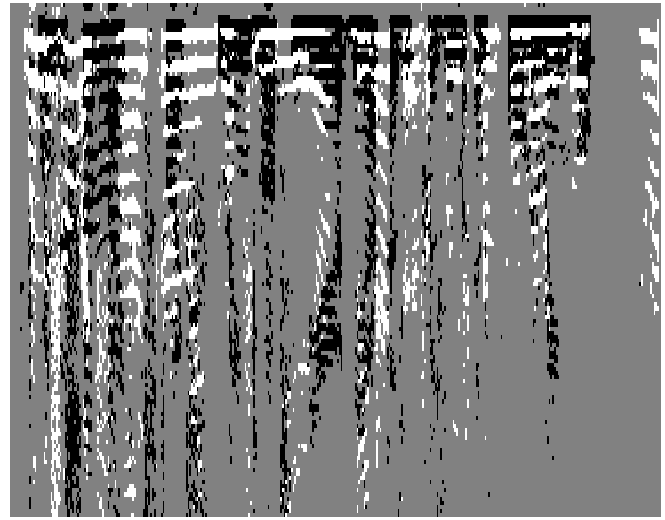


Building training set

Spectrogram of the mix



"Optimal" segmentation



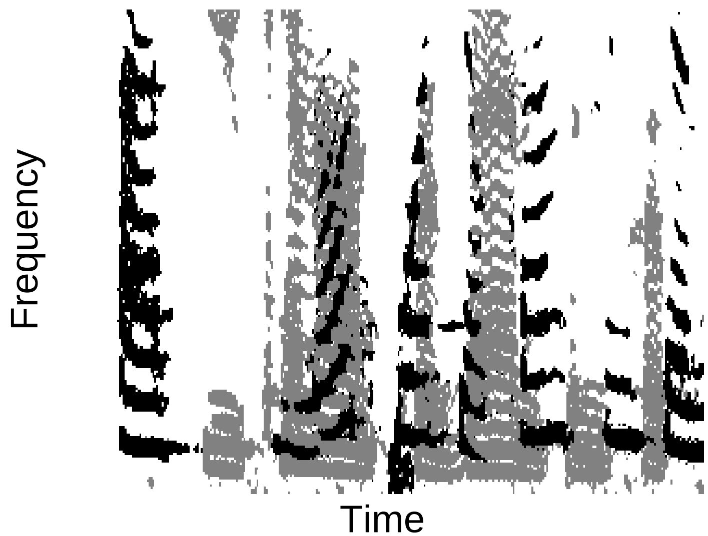
- Empirical property: there exists a segmentation that leads to audibly acceptable signals (e.g., take $\arg\max(|S_1|, |S_2|)$)
- Work as possibly large training datasets
- Requires new way of segmenting images ...
- ... which can be learned from data

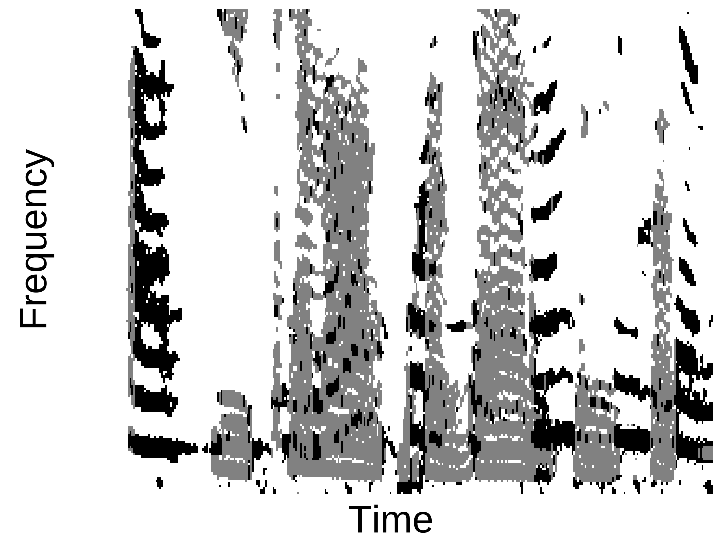
Very large similarity matrices
Linear complexity

- Three different time scales $\Rightarrow W = \alpha_1 W_1 + \alpha_2 W_2 + \alpha_3 W_3$
- **Small**
  - Fine scale structure (continuity, harmonicity)
  - very sparse approximation
- **Medium**
  - Medium scale structure (common fate cues)
  - band-diagonal approximation, potentially reduced rank
- **Large**
  - Global structure (e.g., speaker identification)
  - low-rank approximation (rank is independent of duration)

Experiments

- Two datasets of speakers: one for testing, one for training
- Left: optimal segmentation - right: blind segmentation





- Testing time (Matlab/C): T duration of signal
  - Building features $\approx 4 \times T$
  - Separation $\approx 30 \times T$

Unsupervised learning - Convex relaxations

- **Cuts**: given any matrix $W \in \mathbb{R}^{n \times n}$, find $y \in \{-1,1\}^n$ that minimizes

$$\sum_{i,j=1}^{n} W_{ij} 1_{y_i \neq y_j} = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij}(1 - y_i y_j) = \frac{1}{2} 1^\top W 1 - \frac{1}{2} y^\top W y$$

  – Let $Y = yy^\top$. We have $Y \succcurlyeq 0$, $\mathrm{diag}(Y) = 1$, $\mathrm{rank}(Y) = 1$
  – Convex relaxation (Goemans and Williamson, 1997):

$$\max_{Y \succcurlyeq 0,\ \mathrm{diag}(Y)=1} \mathrm{tr}\, WY$$

  – May be solved as sequence of eigenvalue problems

$$\max_{Y \succcurlyeq 0,\ \mathrm{diag}(Y)=1} \mathrm{tr}\, WY = \min_{\mu \in \mathbb{R}^n} n\lambda_{\max}(W + \mathrm{Diag}(\mu)) - 1^\top \mu$$

Submodular functions

- $F : 2^V \to \mathbb{R}$ is **submodular** if and only if

$$\forall A, B \subset V, \quad F(A) + F(B) \geqslant F(A \cap B) + F(A \cup B)$$

$$\Leftrightarrow \quad \forall k \in V, \quad A \mapsto F(A \cup \{k\}) - F(A) \text{ is non-increasing}$$

Submodular functions

- $F : 2^V \to \mathbb{R}$ is **submodular** if and only if

$$\forall A, B \subset V, \quad F(A) + F(B) \geqslant F(A \cap B) + F(A \cup B)$$

$$\Leftrightarrow \ \forall k \in V, \quad A \mapsto F(A \cup \{k\}) - F(A) \text{ is non-increasing}$$

- **Intuition 1**: defined like concave functions ("diminishing returns")
  - Example: $F : A \mapsto g(\mathrm{Card}(A))$ is submodular if g is concave

Submodular functions

- $F : 2^V \to \mathbb{R}$ is **submodular** if and only if

$$\begin{aligned} & \forall A, B \subset V, \quad F(A) + F(B) \geqslant F(A \cap B) + F(A \cup B) \\ \Leftrightarrow \ & \forall k \in V, \quad A \mapsto F(A \cup \{k\}) - F(A) \text{ is non-increasing} \end{aligned}$$

- **Intuition 1**: defined like concave functions ("diminishing returns")
  - Example: $F : A \mapsto g(\mathrm{Card}(A))$ is submodular if g is concave
- **Intuition 2**: behave like convex functions
  - Polynomial-time minimization, conjugacy theory

Submodular functions

- $F : 2^V \to \mathbb{R}$ is **submodular** if and only if

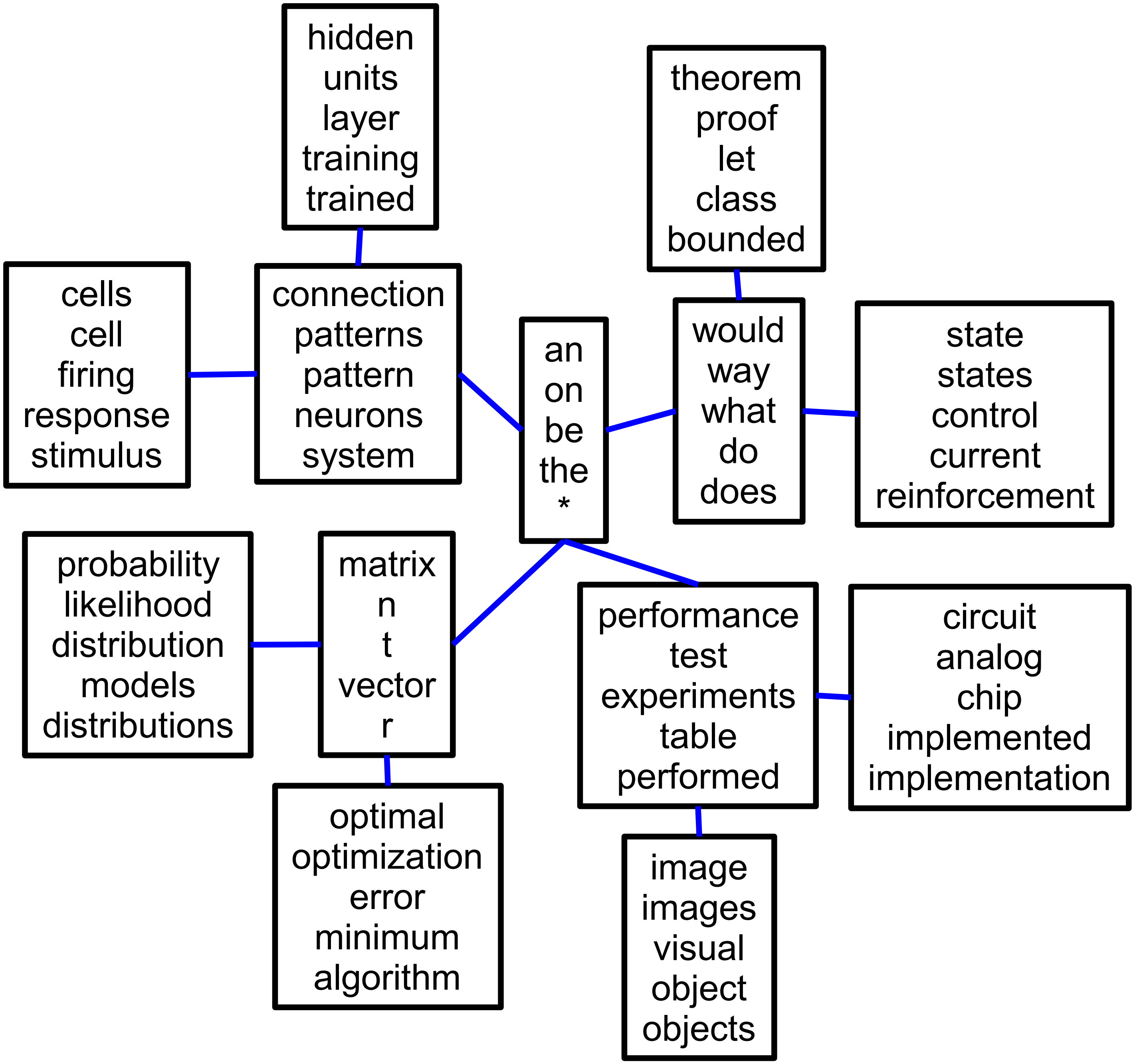
$$\forall A, B \subset V, \quad F(A) + F(B) \geqslant F(A \cap B) + F(A \cup B)$$
$$\Leftrightarrow \quad \forall k \in V, \quad A \mapsto F(A \cup \{k\}) - F(A) \text{ is non-increasing}$$

- **Intuition 1**: defined like concave functions ("diminishing returns")
  - Example: $F : A \mapsto g(\mathrm{Card}(A))$ is submodular if g is concave

- **Intuition 2**: behave like convex functions
  - Polynomial-time minimization, conjugacy theory

- Used in several areas of signal processing and machine learning
  - Total variation/graph cuts
  - Optimal design - **Structured sparsity**

Document modelisation (Jenatton et al., 2010)



Machine Learning and Numerical Analysis
Outline

- **Machine learning**
  - Supervised vs. unsupervised
- **Convex optimization for supervised learning**
  - Sequence of linear systems
- **Spectral methods for unsupervised learning**
  - Sequence of singular value decompositions
- **Combinatorial optimization**
  - Polynomial-time algorithms and convex relaxations

Machine learning - Specificities

- **Low-precision**
  - Objective functions are averages
- **Large scale**
  - Practical impact only when complexity close to linear
- **Online learning**
  - Take advantage of special structure of optimization problems
- **Sequence of problems**
  - Selecting hyperparameters