Kernel methods & sparse methods for computer vision

Francis Bach

Willow project, INRIA - Ecole Normale Supérieure

CVML Summer School, Grenoble, July 2010

Machine learning

- Supervised learning
  - Predict $y \in \mathcal{Y}$ from $x \in \mathcal{X}$, given observations (x_i, y_i), $i = 1, \ldots, n$
- Unsupervised learning
  - Find structure in $x \in \mathcal{X}$, given observations x_i, $i = 1, \ldots, n$
- Application to many problems and data types:
  - **Computer vision**
  - Bioinformatics
  - Text processing
  - etc.
- Specifity: exchanges between **theory** / **algorithms** / **applications**

Machine learning for computer vision

- Multiplication of digital media
- Many different tasks to be solved
  - Associated with different machine learning problems
  - Massive data to learn from

Image retrieval

⇒ Classification, ranking, outlier detection

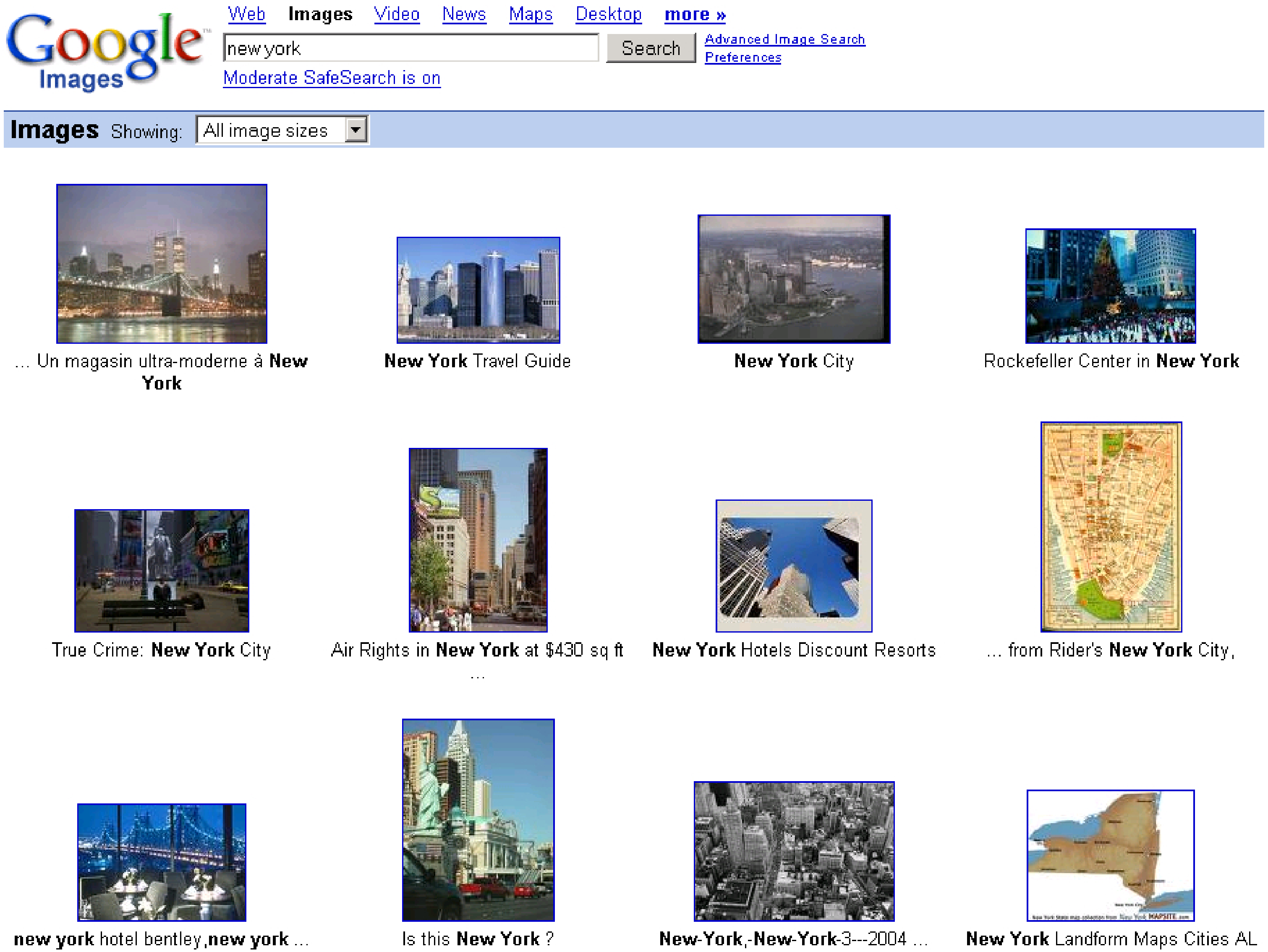


Image retrieval

Classification, ranking, outlier detection

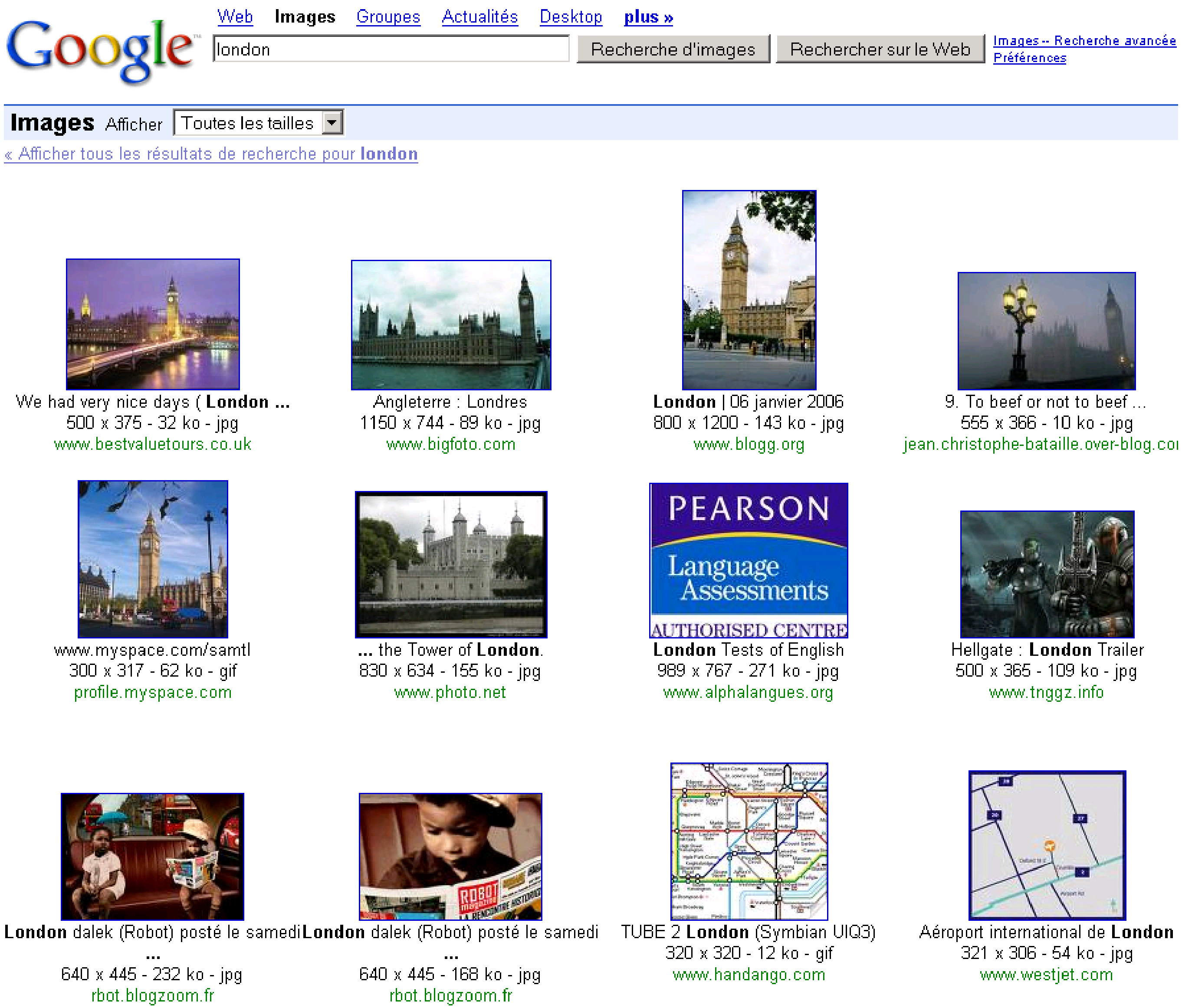


Image retrieval

Classification, ranking, outlier detection

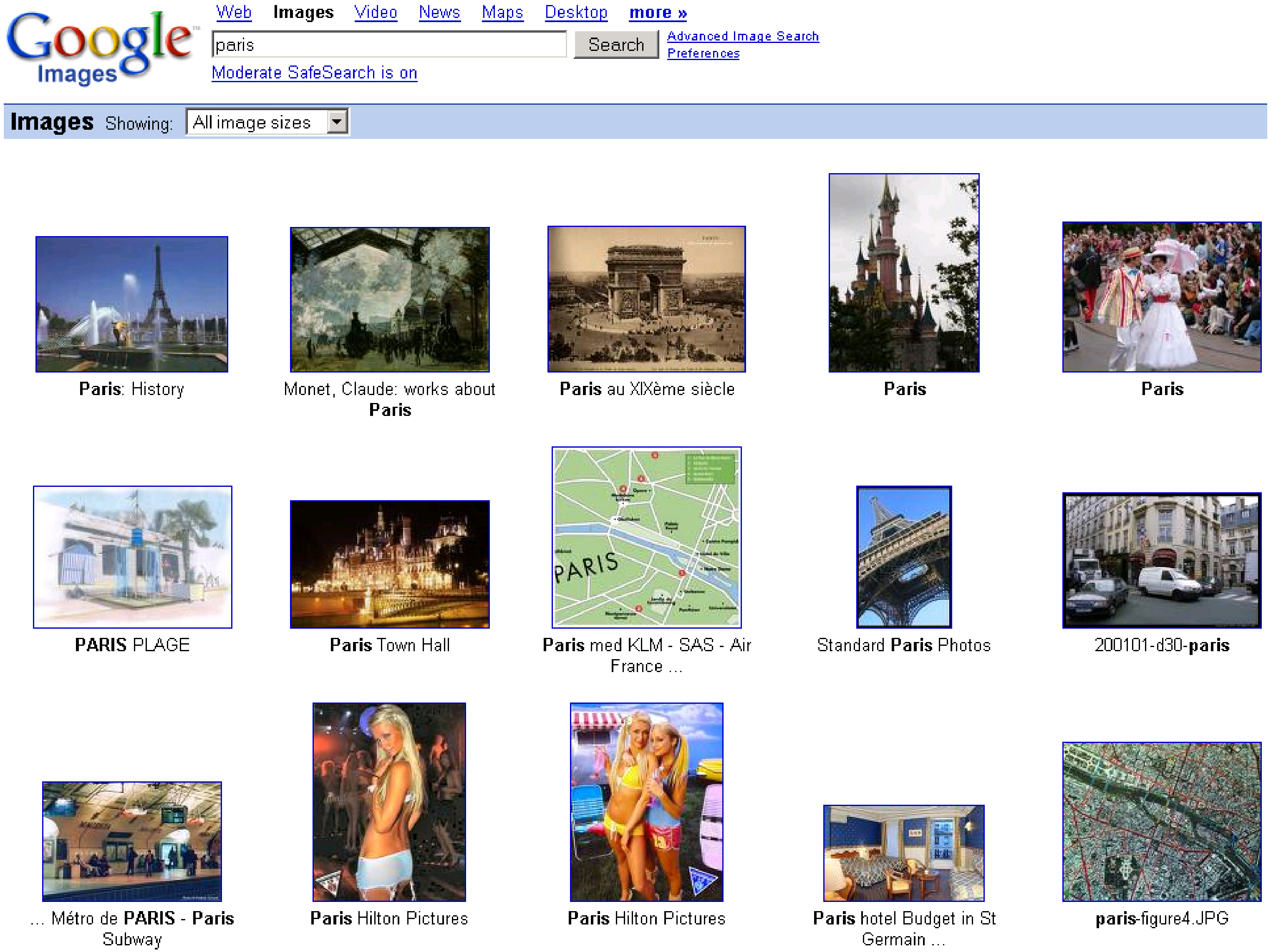


Image annotation
Classification, clustering

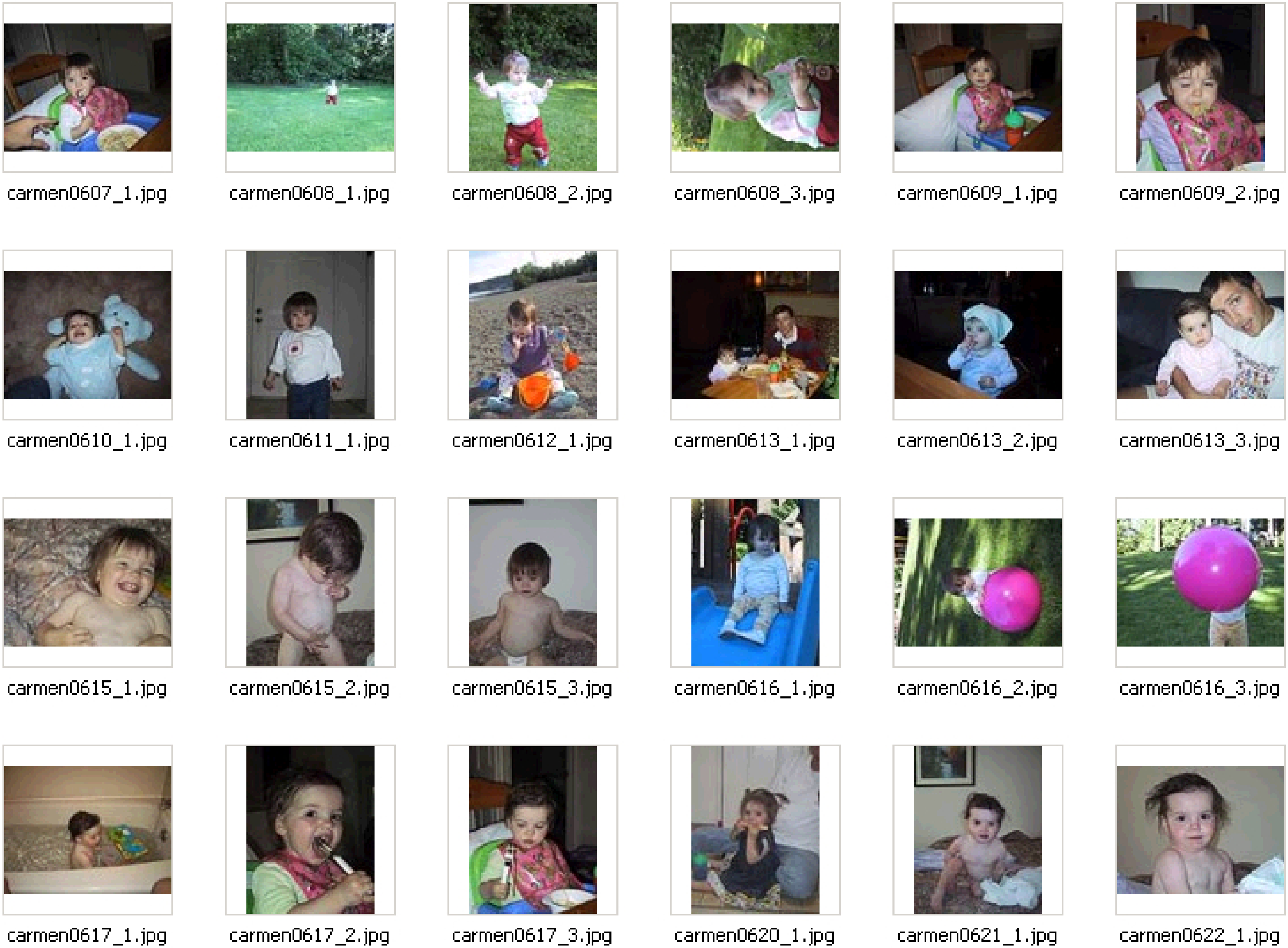





Object recognition ⇒ Multi-label classification

Personal photos

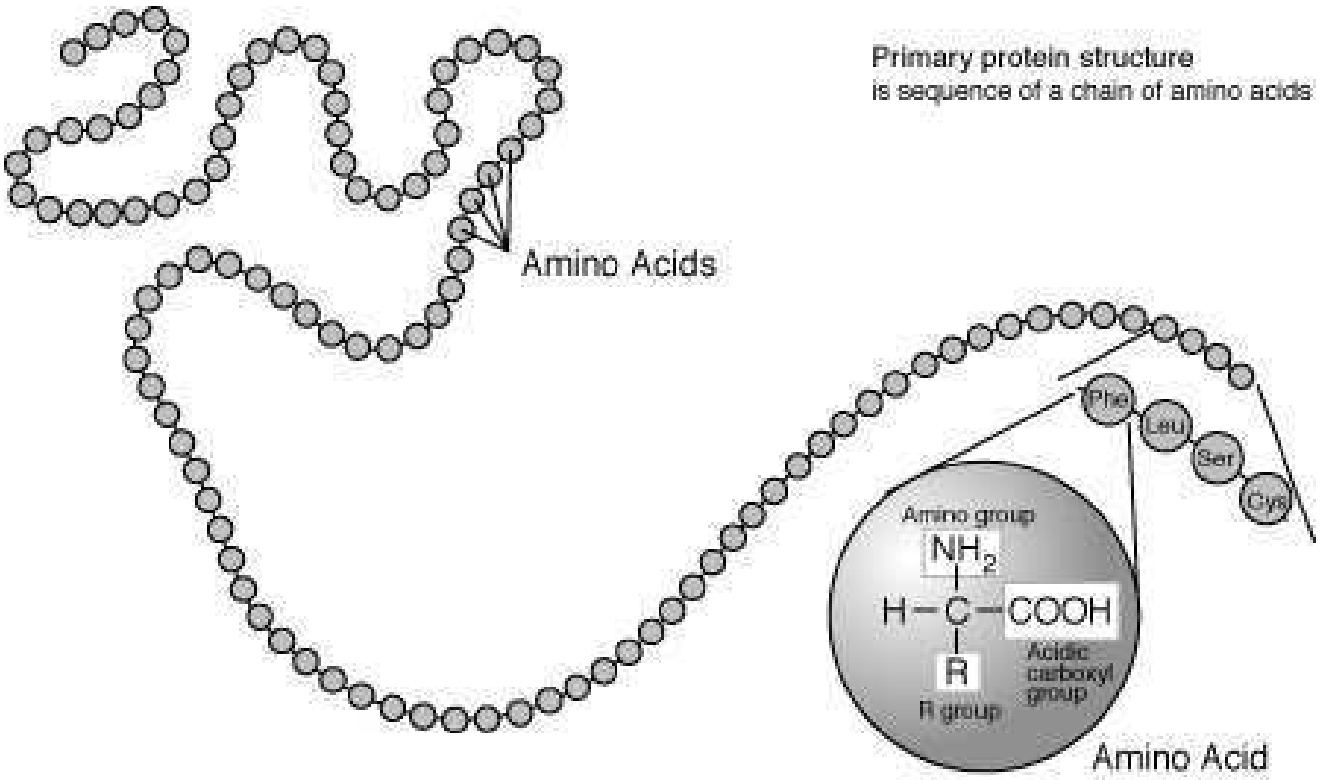
⇒ Classification, clustering, visualization



Machine learning for computer vision

- Multiplication of digital media
- Many different tasks to be solved
  - Associated with different machine learning problems
  - Massive data to learn from
- Similar situations in many fields (e.g., bioinformatics)

Machine learning for bioinformatics (e.g., proteins)



1. Many learning tasks on proteins
   - Classification into functional or structural classes
   - Prediction of cellular localization and interactions

2. Massive data

Machine learning for computer vision

- Multiplication of digital media
- Many different tasks to be solved
  - Associated with different machine learning problems
  - Massive data to learn from
- Similar situations in many fields (e.g., bioinformatics)

  $\Rightarrow$ **Machine learning for high-dimensional data**

Supervised learning and regularization

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$
- Minimize with respect to function $f \in \mathcal{F}$:

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

Error on data + Regularization

Loss & function space ? Norm ?

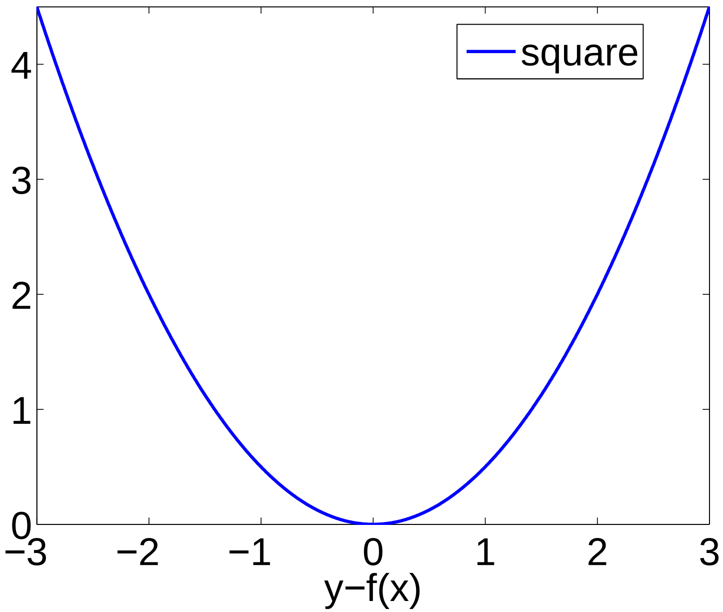
- Two theoretical/algorithmic issues:
  - Loss
  - Function space / norm

Course outline

1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Losses for regression (Shawe-Taylor and Cristianini, 2004)

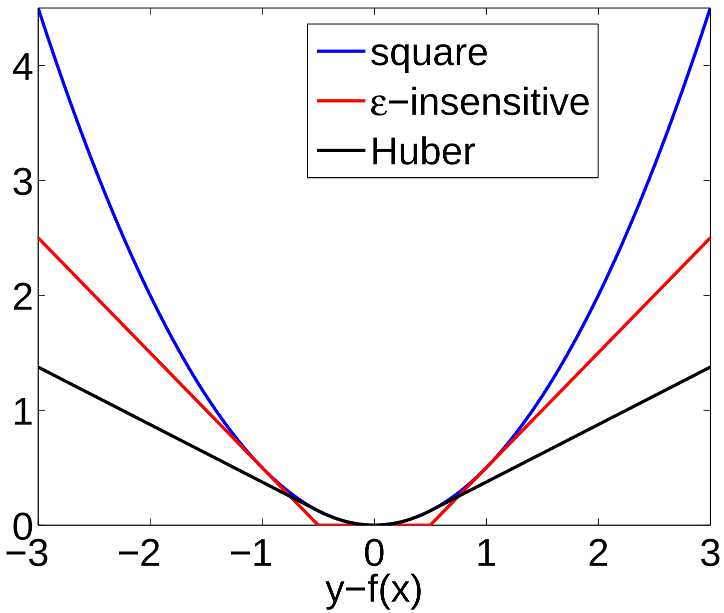
- **Response**: $y \in \mathbb{R}$, prediction $\hat{y} = f(x)$,
  - quadratic (square) loss $\ell(y, f(x)) = \frac{1}{2}(y - f(x))^2$
  - Not many reasons to go beyond square loss!



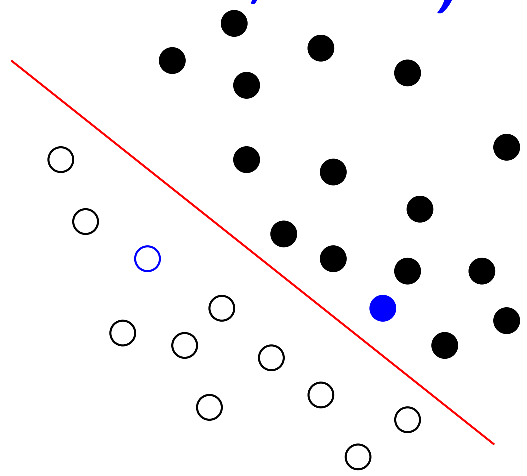
Losses for regression (Shawe-Taylor and Cristianini, 2004)

- **Response**: $y \in \mathbb{R}$, prediction $\hat{y} = f(x)$,
  - quadratic (square) loss $\ell(y, f(x)) = \frac{1}{2}(y - f(x))^2$
  - Not many reasons to go beyond square loss!

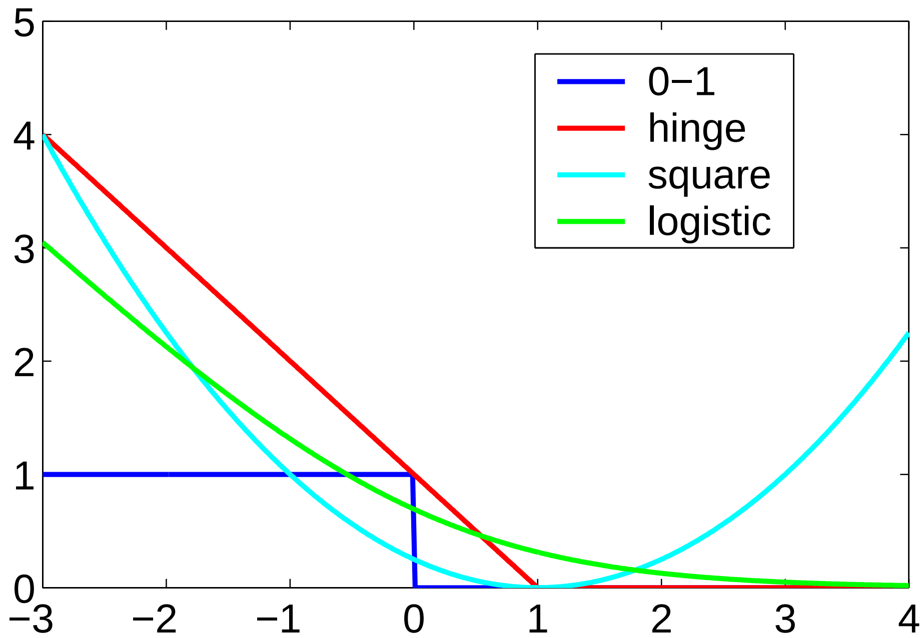
- Other convex losses "with added benefits"
  - ε-insensitive loss $\ell(y, f(x)) = (|y - f(x)| - \varepsilon)_+$
  - Hüber loss (mixed quadratic/linear): robustness to outliers



Losses for classification (Shawe-Taylor and Cristianini, 2004)



- **Label** : $y \in \{-1, 1\}$, 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:



- **Differences between hinge and logistic loss: differentiability/sparsity**

Image annotation $\Rightarrow$ multi-class classification





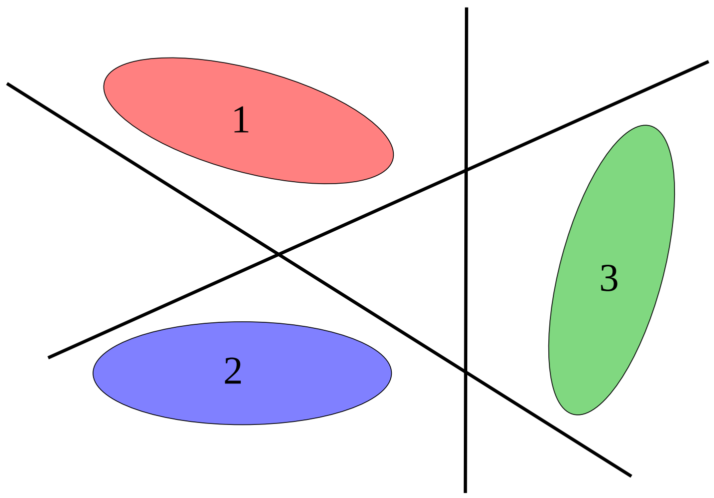

Losses for multi-label classification (Schölkopf and Smola, 2001; Shawe-Taylor and Cristianini, 2004)

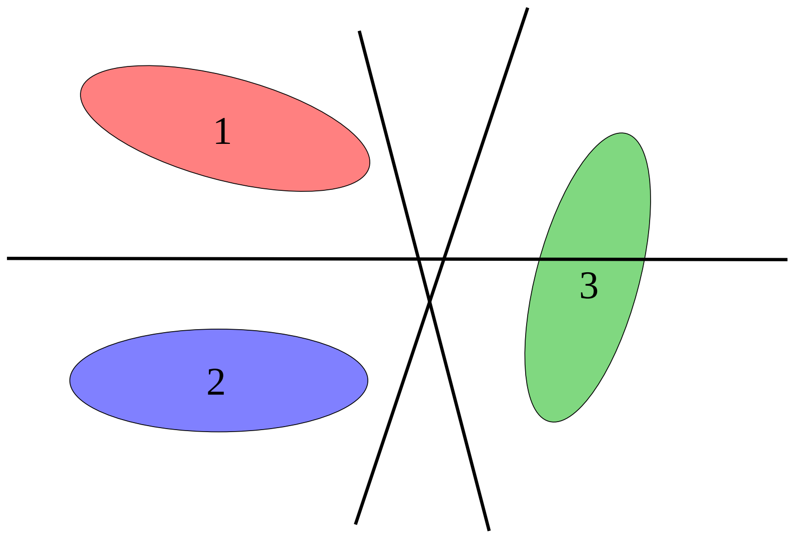
- **Two main strategies** for k classes (with unclear winners)

1. Using existing binary classifiers (efficient code!) $+$ voting schemes
   - "one-vs-rest" : learn k classifiers on the entire data
   - "one-vs-one" : learn $k(k-1)/2$ classifiers on portions of the data

Losses for multi-label classification - Linear predictors

- Using binary classifiers (left: “one-vs-rest”, right: “one-vs-one”)





Losses for multi-label classification (Schölkopf and Smola, 2001; Shawe-Taylor and Cristianini, 2004)

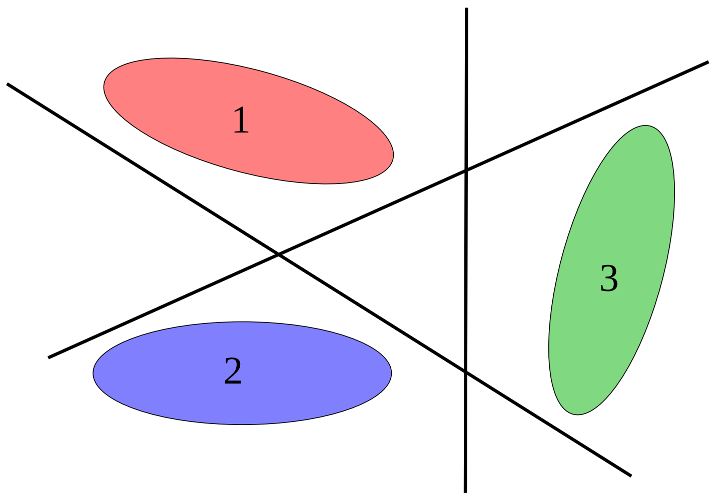
- **Two main strategies** for k classes (with unclear winners)

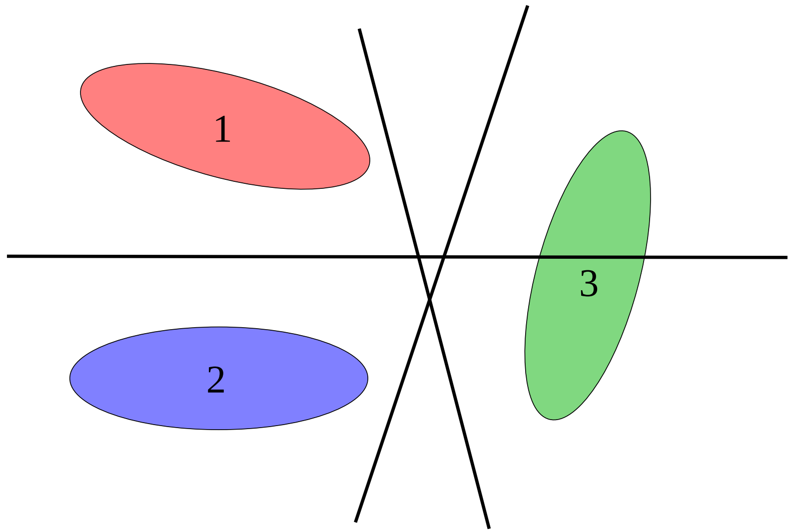
1. Using existing binary classifiers (efficient code!) $+$ voting schemes
   - "one-vs-rest" : learn k classifiers on the entire data
   - "one-vs-one" : learn $k(k-1)/2$ classifiers on portions of the data
2. Dedicated loss functions for prediction using $\arg\max_{i \in \{1,\dots,k\}} f_i(x)$
   - Softmax regression: loss $= -\log(e^{f_y(x)} / \sum_{i=1}^k e^{f_i(x)})$
   - Multi-class SVM - 1: loss $= \sum_{i=1}^k (1 + f_i(x) - f_y(x))_+$
   - Multi-class SVM - 2: loss $= \max_{i \in \{1,\dots,k\}} (1 + f_i(x) - f_y(x))_+$

- Strategies do not consider same predicting functions

Losses for multi-label classification - Linear predictors

- Using binary classifiers (left: "one-vs-rest", right: "one-vs-one")





- Dedicated loss function

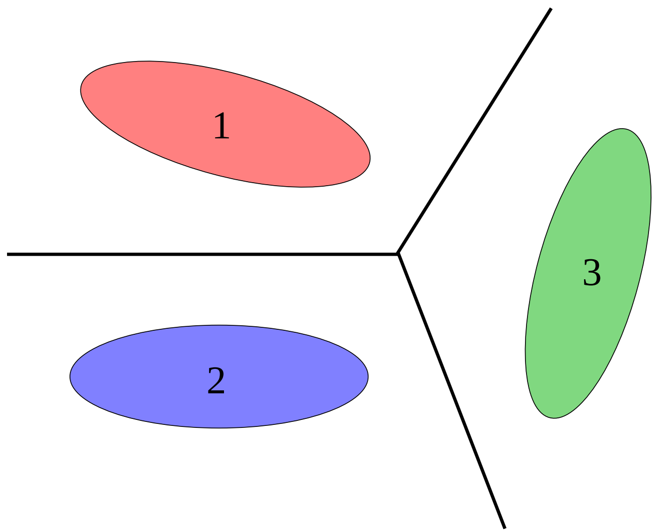


Image retrieval ⇒ ranking

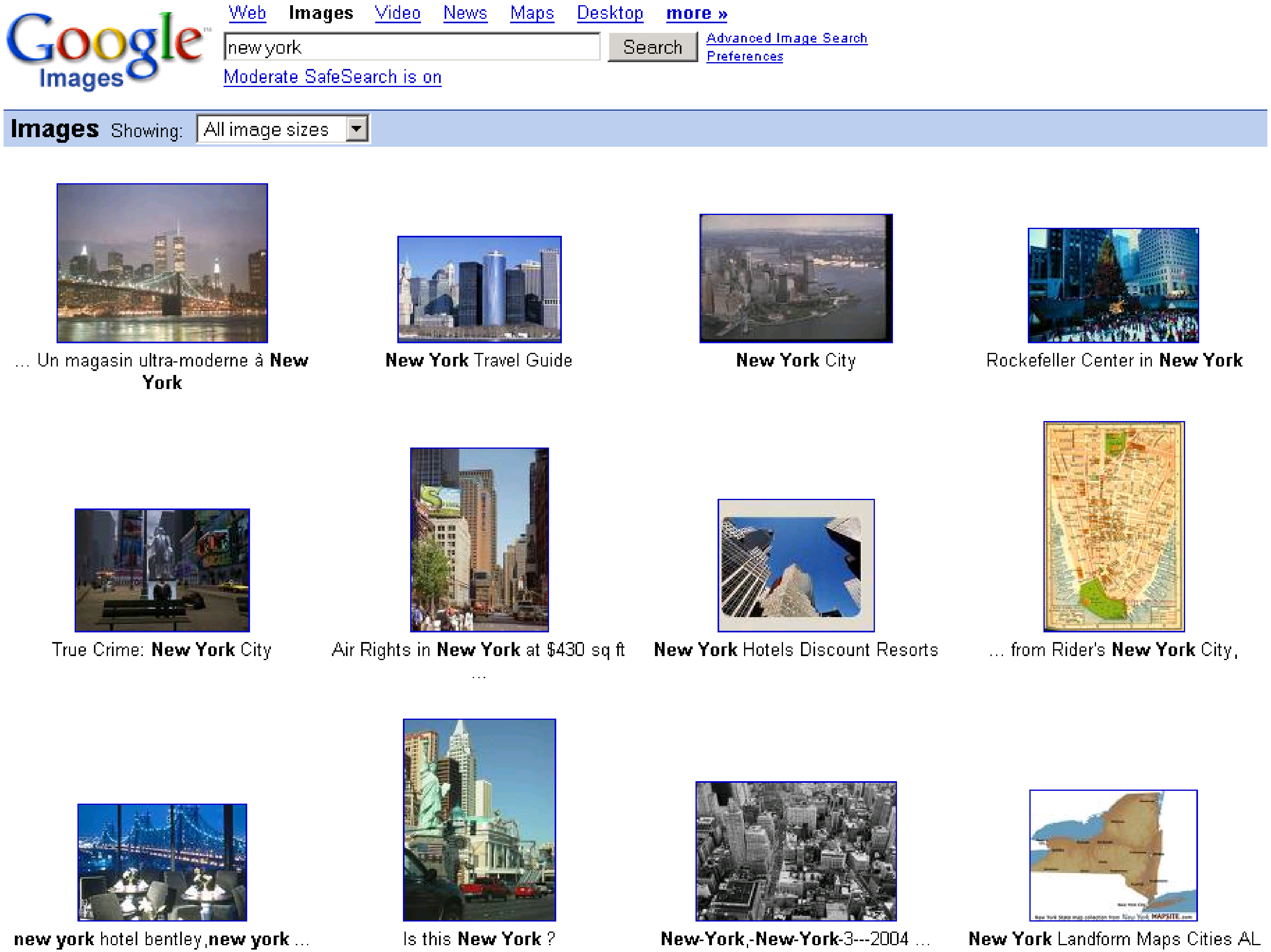
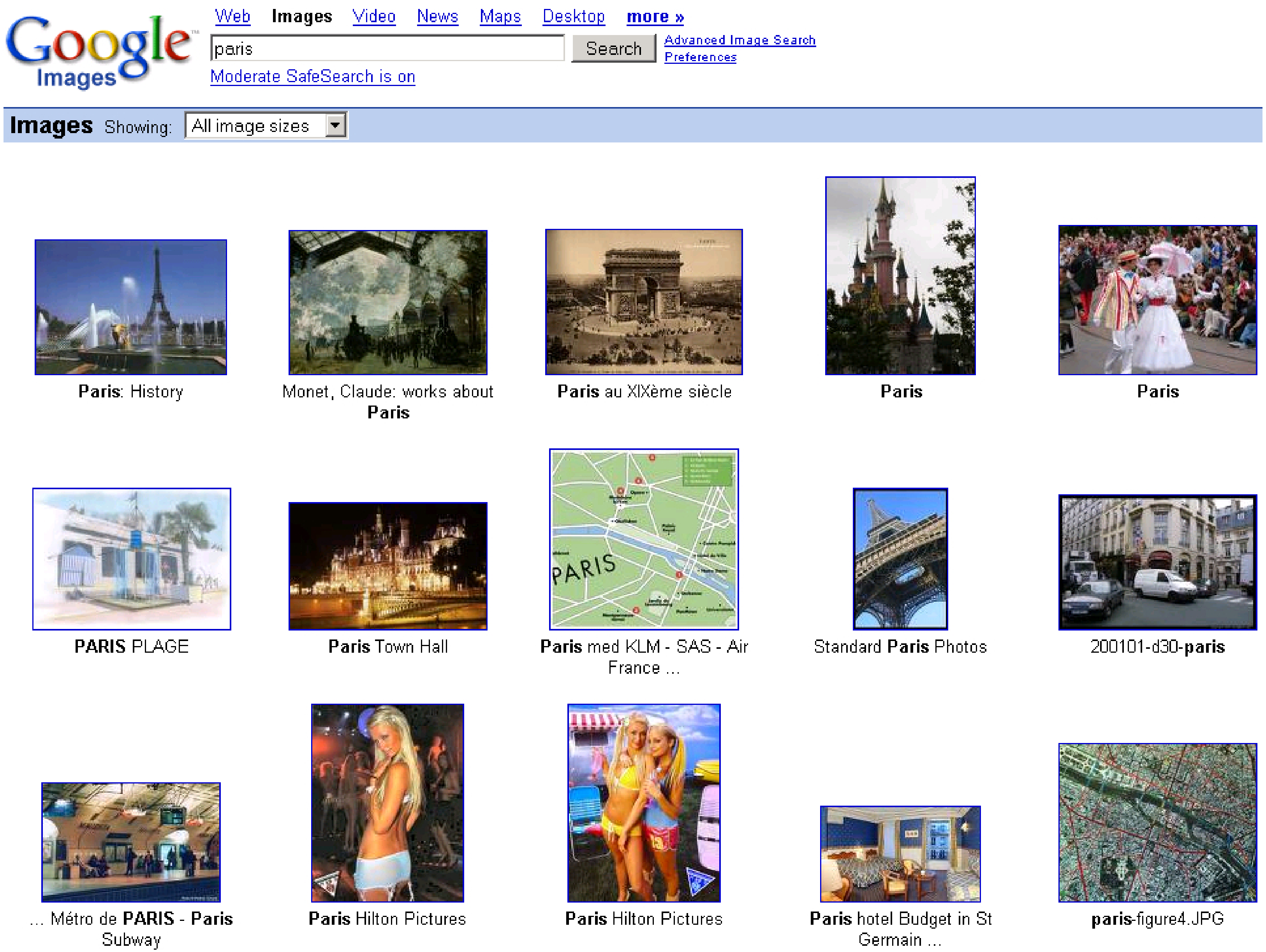


Image retrieval ⇒ outlier/novelty detection



Losses for ther tasks

- Outlier detection (Schölkopf et al., 2001; Vert and Vert, 2006)
  - one-class SVM: learn only with positive examples
- Ranking
  - simple trick: transform into learning on pairs (Herbrich et al., 2000), i.e., predict $\{x > y\}$ or $\{x \leqslant y\}$
  - More general "structured output methods" (Joachims, 2002)
- General structured outputs
  - Very active topic in machine learning and computer vision
  - see, e.g., Taskar (2005)

Dealing with asymmetric cost or unbalanced data in binary classification

- Two cases with similar issues:
  - Asymmetric cost (e.g., spam filterting, detection)
  - Unbalanced data, e.g., lots of positive examples (example: detection)

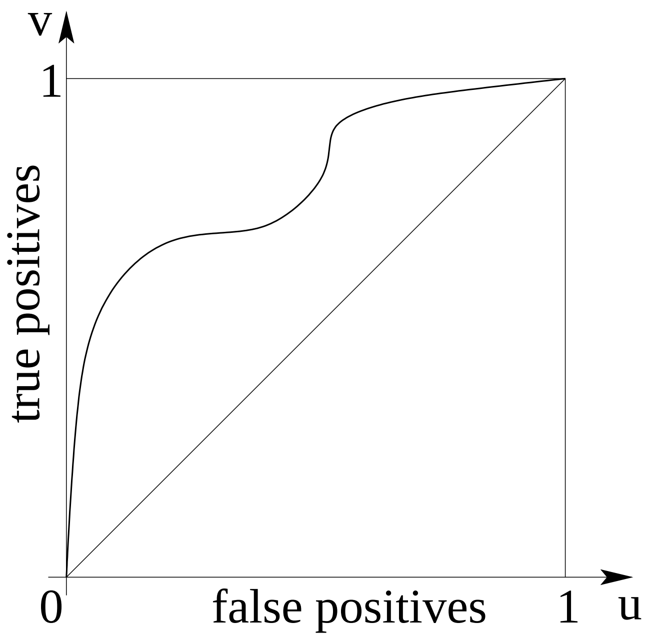
- **One number is not enough to characterize the asymmetric properties**
  - ROC curves (Flach, 2003) – cf. precision-recall curves

- Training using asymmetric losses (Bach et al., 2006)

$$\min_{f \in \mathcal{F}} \quad C_+ \sum_{i, y_i = 1} \ell(y_i f(x_i)) + C_- \sum_{i, y_i = -1} \ell(y_i f(x_i)) + \|f\|^2$$

ROC curves

- ROC plane (u, v)
- $u =$ proportion of false positives $= P(f(x) = 1|y = -1)$
- $v =$ proportion of true positives $= P(f(x) = 1|y = 1)$
- Plot a set of classifiers $f_\gamma(x)$ for $\gamma \in \mathbb{R}$



Course outline

1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Regularizations

- Main goal: avoid overfitting (see course by Jean-Yves Audibert)

- Two main lines of work:

  1. Use Hilbertian (RKHS) norms
     - Non parametric supervised learning and kernel methods
     - Well developped theory (Schölkopf and Smola, 2001; Shawe-Taylor and Cristianini, 2004; Wahba, 1990)
  2. Use "sparsity inducing" norms
     - main example: ℓ_1-norm $\|w\|_1 = \sum_{i=1}^p |w_i|$
     - Perform model selection as well as regularization
     - Theory "in the making"

- **Goal of (this part of) the course: Understand how and when to use these different norms**

Kernel methods for machine learning

- **Definition**: given a set of objects $\mathcal{X}$, a positive definite kernel is a symmetric function $k(x, x')$ such that for all finite sequences of points $x_i \in \mathcal{X}$ and $\alpha_i \in \mathbb{R}$,

$$\sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \geqslant 0$$

  (i.e., the matrix $(k(x_i, x_j))$ is symmetric positive semi-definite)

- Main example: $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$

Kernel methods for machine learning

- **Definition**: given a set of objects $\mathcal{X}$, a positive definite kernel is a symmetric function $k(x,x')$ such that for all finite sequences of points $x_i \in \mathcal{X}$ and $\alpha_i \in \mathbb{R}$,

$$\sum_{i,j} \alpha_i \alpha_j k(x_i,x_j) \geqslant 0$$

(i.e., the matrix $(k(x_i,x_j))$ is symmetric positive semi-definite)

- **Aronszajn theorem** (Aronszajn, 1950): k is a positive definite kernel if and only if there exists a Hilbert space $\mathcal{F}$ and a mapping $\Phi : \mathcal{X} \mapsto \mathcal{F}$ such that

$$\forall (x,x') \in \mathcal{X}^2, \; k(x,x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$$

- $\mathcal{X}$ = "input space", $\mathcal{F}$ = "feature space", Φ = "feature map"

- Functional view: reproducing kernel Hilbert spaces

Classical kernels: kernels on vectors $x \in \mathbb{R}^d$

- Linear kernel $k(x, y) = x^\top y$
  - $\Phi(x) = x$
- Polynomial kernel $k(x, y) = (1 + x^\top y)^d$
  - $\Phi(x) =$ monomials
- Gaussian kernel $k(x, y) = \exp(-\alpha \|x - y\|^2)$
  - $\Phi(x) = ??$
- PROOF

Reproducing kernel Hilbert spaces

- Assume k is a positive definite kernel on $\mathcal{X} \times \mathcal{X}$
- **Aronszajn theorem** (1950): there exists a Hilbert space $\mathcal{F}$ and a mapping $\Phi : \mathcal{X} \mapsto \mathcal{F}$ such that

$$\forall (x, x') \in \mathcal{X}^2, \ \ k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$$

- $\mathcal{X}$ = "input space", $\mathcal{F}$ = "feature space", Φ = "feature map"
- RKHS: particular instantiation of $\mathcal{F}$ as a function space
  - $\Phi(x) = k(\cdot, x)$
  - function evaluation $\boxed{f(x) = \langle f, \Phi(x) \rangle}$
  - reproducing property: $k(x, y) = \langle k(\cdot, x), k(\cdot, y) \rangle$
- Notations : $f(x) = \langle f, \Phi(x) \rangle = f^\top \Phi(x)$, $\|f\|^2 = \langle f, f \rangle$

Classical kernels: kernels on vectors $x \in \mathbb{R}^d$

- Linear kernel $k(x,y) = x^\top y$
  - Linear functions
- Polynomial kernel $k(x,y) = (1 + x^\top y)^d$
  - Polynomial functions
- Gaussian kernel $k(x,y) = \exp(-\alpha \|x-y\|^2)$
  - Smooth functions

Classical kernels: kernels on vectors $x \in \mathbb{R}^d$

- Linear kernel $k(x, y) = x^\top y$
  - Linear functions
- Polynomial kernel $k(x, y) = (1 + x^\top y)^d$
  - Polynomial functions
- Gaussian kernel $k(x, y) = \exp(-\alpha \|x - y\|^2)$
  - Smooth functions
- **Parameter selection? Structured domain?**

Regularization and representer theorem

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$, kernel k (with RKHS $\mathcal{F}$)

- Minimize with respect to f: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^n \ell(y_i, f^\top \Phi(x_i)) + \frac{\lambda}{2} \|f\|^2}$

- No assumptions on cost ℓ or n

- **Representer theorem** (Kimeldorf and Wahba, 1971): optimum is reached for weights of the form
$$f = \sum_{j=1}^n \alpha_j \Phi(x_j) = \sum_{j=1}^n \alpha_j k(\cdot, x_j)$$

- PROOF

Regularization and representer theorem

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$, kernel k (with RKHS $\mathcal{F}$)
- Minimize with respect to f: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^n \ell(y_i, f^\top \Phi(x_i)) + \frac{\lambda}{2} \|f\|^2}$
- No assumptions on cost ℓ or n
- **Representer theorem** (Kimeldorf and Wahba, 1971): optimum is reached for weights of the form
$$f = \sum_{j=1}^n \alpha_j \Phi(x_j) = \sum_{j=1}^n \alpha_j k(\cdot, x_j)$$
- $\alpha \in \mathbb{R}^n$ dual parameters, $K \in \mathbb{R}^{n \times n}$ kernel matrix:
$$K_{ij} = \Phi(x_i)^\top \Phi(x_j) = k(x_i, x_j)$$
- Equivalent problem: $\boxed{\min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i) + \frac{\lambda}{2} \alpha^\top K \alpha}$

Kernel trick and modularity

- Kernel trick: any algorithm for finite-dimensional vectors that only uses pairwise dot-products can be applied in the feature space.
  - Replacing dot-products by kernel functions
  - Implicit use of (very) large feature spaces
  - Linear to non-linear learning methods

Kernel trick and modularity

- Kernel trick: any algorithm for finite-dimensional vectors that only uses pairwise dot-products can be applied in the feature space.
  - Replacing dot-products by kernel functions
  - Implicit use of (very) large feature spaces
  - Linear to non-linear learning methods

- Modularity of kernel methods
  1. Work on new algorithms and theoretical analysis
  2. Work on new kernels for specific data types

Representer theorem and convex duality

- The parameters $\alpha \in \mathbb{R}^n$ may also be interpreted as Lagrange multipliers

- Assumption: cost function is convex, $\varphi_i(u_i) = \ell(y_i, u_i)$

- Primal problem: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^n \varphi_i(f^\top \Phi(x_i)) + \frac{\lambda}{2}\|f\|^2}$

- What about the constant term b? replace $\Phi(x)$ by $(\Phi(x), c)$, c large

	$\varphi_i(u_i)$
LS regression	$\frac{1}{2}(y_i - u_i)^2$
Logistic regression	$\log(1 + \exp(-y_i u_i))$
SVM	$(1 - y_i u_i)_+$

Representer theorem and convex duality
Proof

- Primal problem: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^n \varphi_i(f^\top \Phi(x_i)) + \frac{\lambda}{2}\|f\|^2}$

- Define $\psi_i(v_i) = \max_{u_i \in \mathbb{R}} v_i u_i - \varphi_i(u_i)$ as the *Fenchel conjugate* of φ_i

- Main trick: introduce constraint $u_i = f^\top \Phi(x_i)$ and associated Lagrange multipliers α_i

- Lagrangian $\mathcal{L}(\alpha, f) = \displaystyle\sum_{i=1}^n \varphi_i(u_i) + \frac{\lambda}{2}\|f\|^2 + \lambda \sum_{i=1}^n \alpha_i(u_i - f^\top \Phi(x_i))$

  – Maximize with respect to $u_i \Rightarrow$ term of the form $-\psi_i(-\lambda \alpha_i)$
  – Maximize with respect to $f \Rightarrow f = \sum_{i=1}^n \alpha_i \Phi(x_i)$

Representer theorem and convex duality

- Assumption: cost function is convex $\varphi_i(u_i) = \ell(y_i, u_i)$
- Primal problem: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^n \varphi_i(f^\top \Phi(x_i)) + \frac{\lambda}{2} \|f\|^2}$
- Dual problem: $\boxed{\max_{\alpha \in \mathbb{R}^n} - \sum_{i=1}^n \psi_i(-\lambda \alpha_i) - \frac{\lambda}{2} \alpha^\top K \alpha}$

  where $\psi_i(v_i) = \max_{u_i \in \mathbb{R}} v_i u_i - \varphi_i(u_i)$ is the Fenchel conjugate of φ_i
- Strong duality
- Relationship between primal and dual variables (at optimum):
$$f = \sum_{i=1}^n \alpha_i \Phi(x_i)$$
- NB: adding constant term $b \Leftrightarrow$ add constraints $\sum_{i=1}^n \alpha_i = 0$

“Classical” kernel learning (2-norm regularization)

Primal problem	$\min_{f \in \mathcal{F}} \left(\sum_i \varphi_i(f^\top \Phi(x_i)) + \frac{\lambda}{2} \|\|f\|\|^2 \right)$
Dual problem	$\max_{\alpha \in \mathbb{R}^n} \left(-\sum_i \psi_i(\lambda \alpha_i) - \frac{\lambda}{2} \alpha^\top K \alpha \right)$
Optimality conditions	$f = \sum_{i=1}^n \alpha_i \Phi(x_i)$

- Assumptions on loss φ_i:
  - $\varphi_i(u)$ convex
  - $\psi_i(v)$ Fenchel conjugate of $\varphi_i(u)$, i.e., $\psi_i(v) = \max_{u \in \mathbb{R}} (vu - \varphi_i(u))$

	$\varphi_i(u_i)$	$\psi_i(v)$
LS regression	$\frac{1}{2}(y_i - u_i)^2$	$\frac{1}{2}v^2 + vy_i$
Logistic regression	$\log(1 + \exp(-y_i u_i))$	$(1+vy_i)\log(1+vy_i) - vy_i \log(-vy_i)$
SVM	$(1 - y_i u_i)_+$	$vy_i \times 1_{-vy_i \in [0,1]}$

Particular case of the support vector machine

- Primal problem: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^n (1 - y_i f^\top \Phi(x_i))_+ + \frac{\lambda}{2} \|f\|^2}$

- Dual problem: $\boxed{\max_{\alpha \in \mathbb{R}^n} \left(- \sum_i \lambda \alpha_i y_i \times 1_{-\lambda \alpha_i y_i \in [0,1]} - \frac{\lambda}{2} \alpha^\top K \alpha \right)}$

- Dual problem (by change of variable $\alpha \leftarrow -\operatorname{Diag}(y)\alpha$ and $C = 1/\lambda$):

$$\boxed{\max_{\alpha \in \mathbb{R}^n,\ 0 \leqslant \alpha \leqslant C} \sum_{i=1}^n \alpha_i - \frac{1}{2} \alpha^\top \operatorname{Diag}(y) K \operatorname{Diag}(y) \alpha}$$

Particular case of the support vector machine

- Primal problem: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (1 - y_i f^\top \Phi(x_i))_+ + \frac{\lambda}{2} \|f\|^2}$

- Dual problem:

$$\boxed{\max_{\alpha \in \mathbb{R}^n,\ 0 \leqslant \alpha \leqslant C} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \alpha^\top \operatorname{Diag}(y) K \operatorname{Diag}(y) \alpha}$$

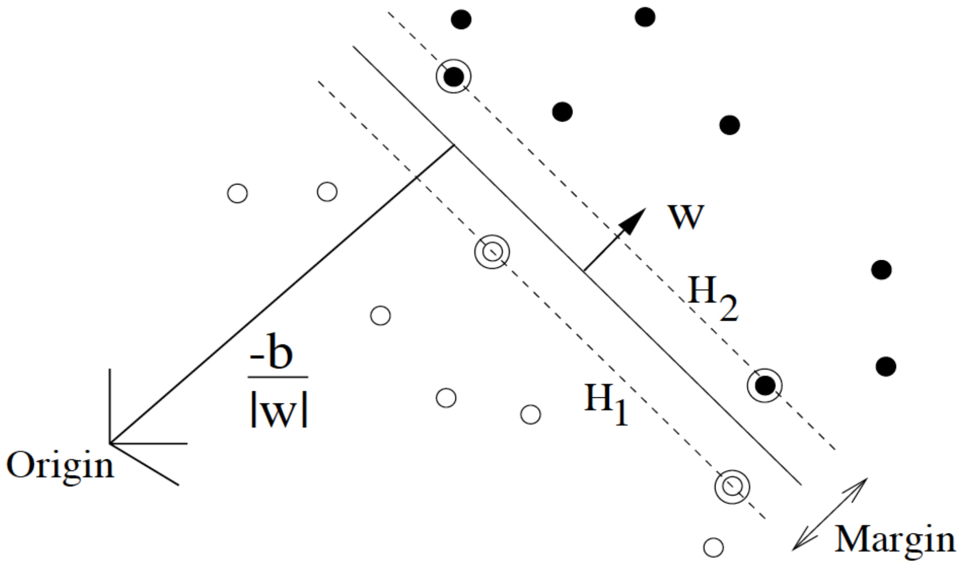
Particular case of the support vector machine

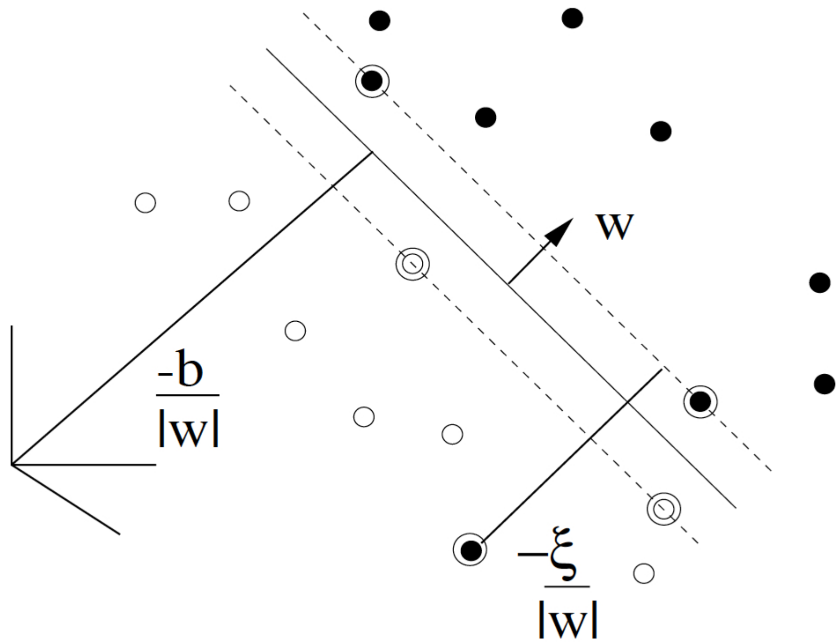
- Primal problem: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (1 - y_i f^\top \Phi(x_i))_+ + \frac{\lambda}{2} \|f\|^2}$

- Dual problem:

$$\boxed{\max_{\alpha \in \mathbb{R}^n,\ 0 \leqslant \alpha \leqslant C} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \alpha^\top \mathrm{Diag}(y) K \,\mathrm{Diag}(y) \alpha}$$

- What about the traditional picture?





Course outline

1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Kernel ridge regression (a.k.a spline smoothing) - I

- Data $x_1, \dots, x_n \in \mathcal{X}$, p.d. kernl k, $y_1, \dots, y_n \in \mathbb{R}$

- Least-squares

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{F}}^2$$

- View 1: representer theorem $\Rightarrow f = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$
  - equivalent to

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} (y_i - (K\alpha)_i)^2 + \lambda \alpha^\top K \alpha$$

  - Solution equal to $\alpha = (K + n\lambda I)^{-1} y + \varepsilon$ with $K\varepsilon = 0$
  - Unique solution f

Kernel ridge regression (a.k.a spline smoothing) - II

- Links with spline smoothing

- Other view: $\mathcal{F} \in \mathbb{R}^d$, $\Phi \in \mathbb{R}^{n \times d}$

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \|y - \Phi w\|^2 + \lambda \|w\|^2$$

- Solution equal to $w = (\Phi^\top \Phi + n\lambda I)^{-1} \Phi^\top y$

- Note that $w = \Phi^\top (\Phi \Phi^\top + n\lambda I)^{-1} y$

- Φw equal to $K\alpha$

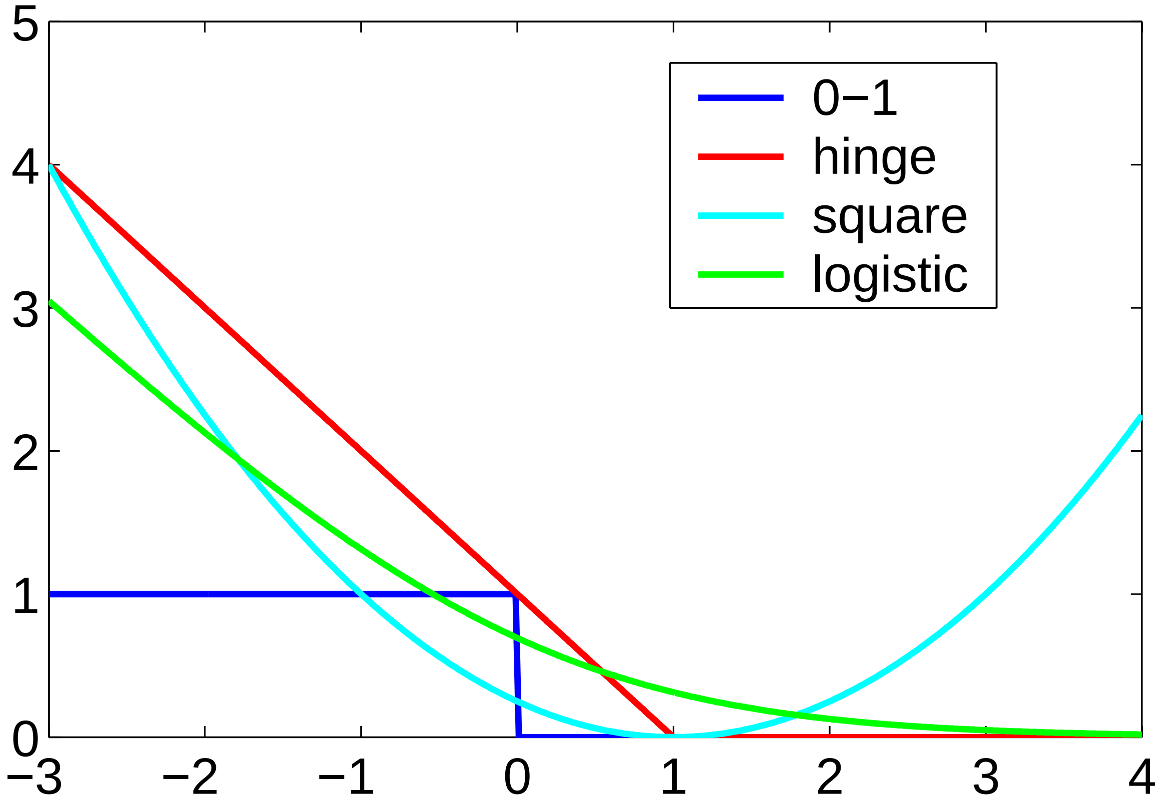
Kernel ridge regression (a.k.a spline smoothing) - III

- Dual view:
  - dual problem: $\max_{\alpha \in \mathbb{R}^n} -\frac{n\lambda}{2}\|\alpha\|^2 - \alpha^\top y - \frac{1}{2}\alpha^\top K \alpha$
  - solution: $\alpha = (K + \lambda I)^{-1} y$

- Warning: same solution obtained from different point of views

Losses for classification

- Usual convex costs:



- **Differences between hinge and logistic loss: differentiability/sparsity**

Support vector machine or logistic regression?

- Predictive performance is similar
- Only true difference is numerical
  - SVM: sparsity in α
  - Logistic: differentiable loss function
- Which one to use?
  - Linear kernel $\Rightarrow$ Logistic $+$ Newton/Gradient descent
  - Nonlinear kernel $\Rightarrow$ SVM $+$ dual methods or simpleSVM

Algorithms for supervised kernel methods

- Four formulations
  1. Dual: $\max_{\alpha \in \mathbb{R}^n} - \sum_i \psi_i(\lambda \alpha_i) - \frac{\lambda}{2} \alpha^\top K \alpha$
  2. Primal: $\min_{f \in \mathcal{F}} \sum_i \varphi_i(f^\top \Phi(x_i)) + \frac{\lambda}{2} ||f||^2$
  3. Primal + Representer: $\min_{\alpha \in \mathbb{R}^n} \sum_i \varphi_i((K\alpha)_i) + \frac{\lambda}{2} \alpha^\top K \alpha$
  4. Convex programming

- **Best strategy depends on loss (differentiable or not) and kernel (linear or not)**

Dual methods

- Dual problem: $\max_{\alpha \in \mathbb{R}^n} - \sum_i \psi_i(\lambda \alpha_i) - \frac{\lambda}{2} \alpha^\top K \alpha$

- Main method: coordinate descent (a.k.a. sequential minimal optimization - SMO) (Platt, 1998; Bottou and Lin, 2007; Joachims, 1998)
  - Efficient when loss is piecewise quadratic (i.e., hinge = SVM)
  - Sparsity may be used in the case of the SVM

- Computational complexity: between quadratic and cubic in n

- **Works for all kernels**

Primal methods

- Primal problem: $\min_{f \in \mathcal{F}} \sum_i \varphi_i(f^\top \Phi(x_i)) + \frac{\lambda}{2} ||f||^2$

- Only works directly if $\Phi(x)$ may be built explicitly and has small dimension
  - Example: linear kernel in small dimensions

- Differentiable loss: gradient descent or Newton's method are very efficient in small dimensions

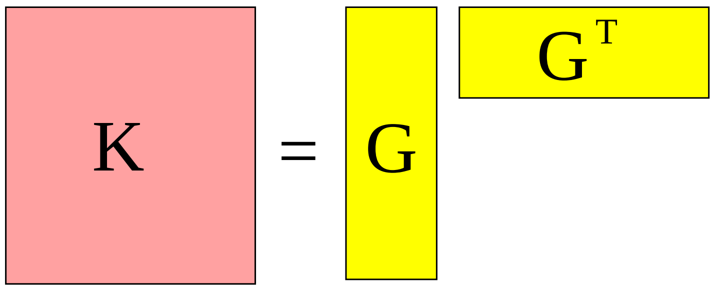
- Larger scale: stochastic gradient descent (Shalev-Shwartz et al., 2007; Bottou and Bousquet, 2008)

Primal methods with representer theorems

- Primal problem in α: $\min_{\alpha \in \mathbb{R}^n} \sum_i \varphi_i((K\alpha)_i) + \frac{\lambda}{2} \alpha^\top K \alpha$

- Direct optimization in α poorly conditioned (K has low-rank) unless Newton method is used (Chapelle, 2007)

- General kernels: use incomplete Cholesky decomposition (Fine and Scheinberg, 2001; Bach and Jordan, 2002) to obtain a square root $K = GG^\top$



G of size $n \times m$,
where $m \ll n$

  - "Empirical input space" of size m obtained using rows of G
  - Running time to compute G: $O(m^2 n)$

Direct convex programming

- **Convex programming toolboxes $\Rightarrow$ very inefficient!**
- May use special structure of the problem
  - e.g., SVM and sparsity in α
- Active set method for the SVM: **SimpleSVM** (Vishwanathan et al., 2003; Loosli et al., 2005)
  - Cubic complexity in the number of support vectors
- Full regularization path for the SVM (Hastie et al., 2005; Bach et al., 2006)
  - Cubic complexity in the number of support vectors
  - May be extended to other settings (Rosset and Zhu, 2007)

Course outline

1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Kernel methods - I

- Distances in the "feature space"

$$d_k(x,y)^2 = \|\Phi(x) - \Phi(y)\|_{\mathcal{F}}^2 = k(x,x) + k(y,y) - 2k(x,y)$$

- Nearest-neighbor classification/regression

Kernel methods - II
Simple discrimination algorithm

- Data $x_1, \ldots, x_n \in \mathcal{X}$, classes $y_1, \ldots, y_n \in \{-1, 1\}$
- Compare distances to mean of each class
- Equivalent to classifying x using the sign of

$$\frac{1}{\#\{i, y_i = 1\}} \sum_{i, y_i = 1} k(x, x_i) - \frac{1}{\#\{i, y_i = -1\}} \sum_{i, y_i = -1} k(x, x_i)$$

- Proof...
- Geometric interpretation of Parzen windows

Kernel methods - III
Data centering

- n points $x_1, \ldots, x_n \in \mathcal{X}$
- kernel matrix $K \in \mathbb{R}^n$, $K_{ij} = k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$
- Kernel matrix of centered data $\tilde{K}_{ij} = \langle \Phi(x_i) - \mu, \Phi(x_j) - \mu \rangle$ where $\mu = \frac{1}{n} \sum_{i=1}^n \Phi(x_i)$
- Formula: $\tilde{K} = \Pi_n K \Pi_n$ with $\Pi_n = I_n - \frac{E}{n}$, and E constant matrix equal to 1.
- Proof...
- NB: μ is not of the form $\Phi(z)$, $z \in \mathcal{X}$ (cf. preimage problem)

Kernel PCA

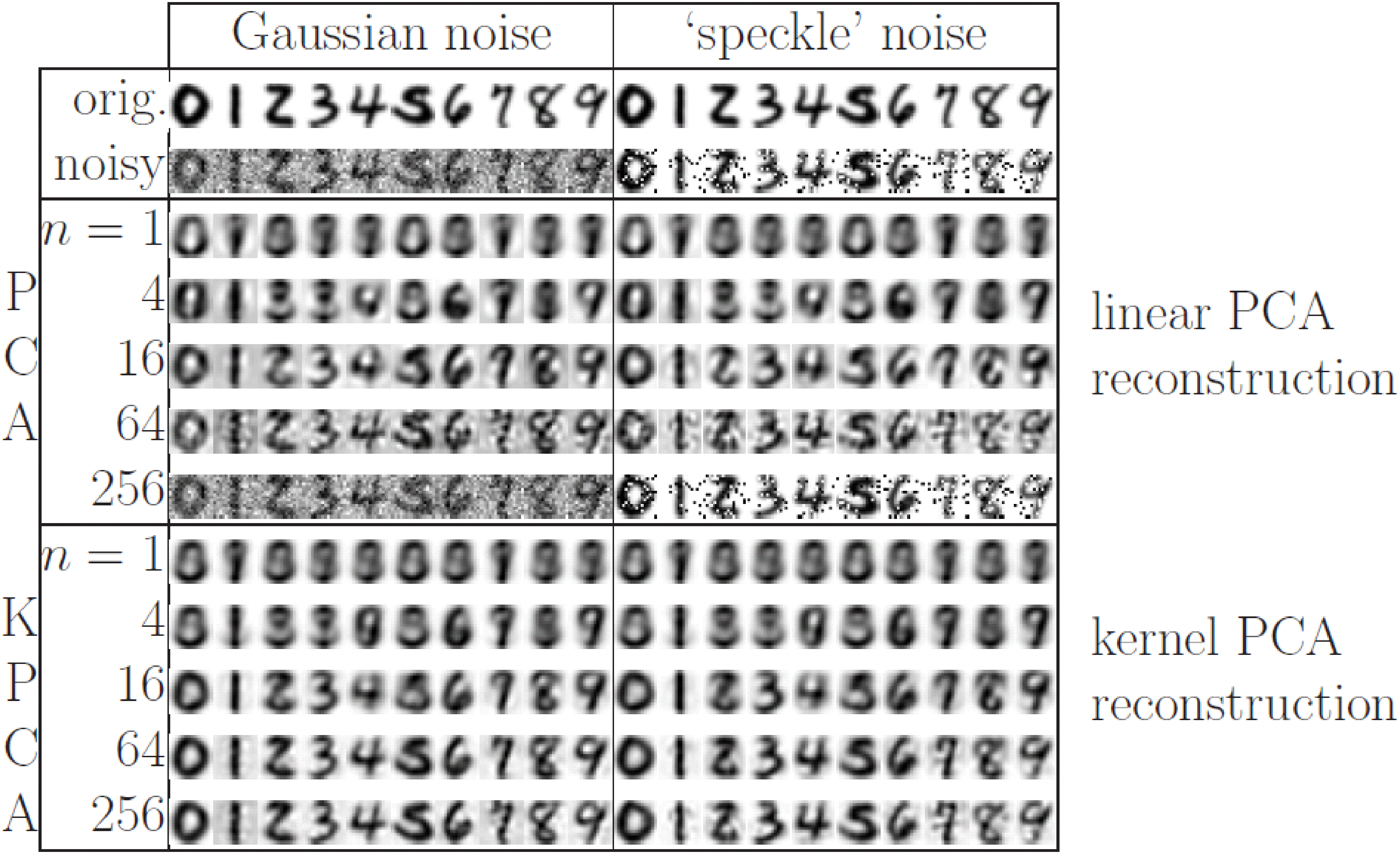
- Linear principal component analysis
  - data $x_1, \dots, x_n \in \mathbb{R}^p$,

$$\max_{w \in \mathbb{R}^p} \frac{w^\top \hat{\Sigma} w}{w^\top w} = \max_{w \in \mathbb{R}^p} \frac{\operatorname{var}(w^\top X)}{w^\top w}$$

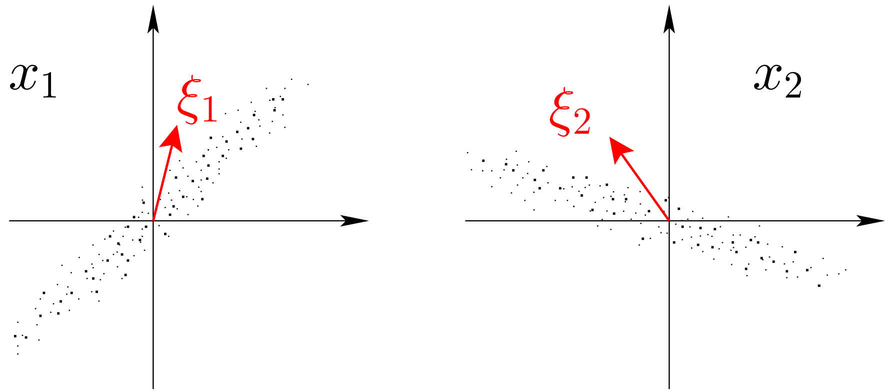
  - w is largest eigenvector of $\hat{\Sigma}$
  - Denoising, data representation

- Kernel PCA: data $x_1, \dots, x_n \in \mathcal{X}$, p.d. kernel k
  - View 1: $\max_{w \in \mathcal{F}} \frac{\operatorname{var}(\langle \Phi(X), w \rangle)}{w^\top w}$ View 2: $\max_{f \in \mathcal{F}} \frac{\operatorname{var}(f(X))}{\|f\|_{\mathcal{F}}^2}$
  - Solution: $f, w = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$ and α first eigenvector of $\tilde{K} = \Pi_n K \Pi_n$
  - Interpretation in terms of covariance operators

Denoising with kernel PCA (From Schölkopf, 2005)



Canonical correlation analysis



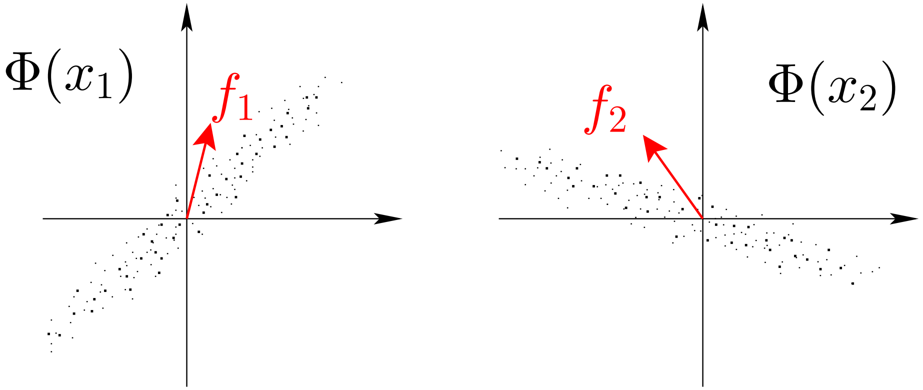
- Given two multivariate random variables x_1 and x_2, finds the pair of directions ξ_1, ξ_2 with maximum correlation:

$$\rho(x_1, x_2) = \max_{\xi_1, \xi_2} \operatorname{corr}(\xi_1^T x_1, \xi_2^T x_2) = \max_{\xi_1, \xi_2} \frac{\xi_1^T C_{12} \xi_2}{\left(\xi_1^T C_{11} \xi_1\right)^{1/2} \left(\xi_2^T C_{22} \xi_2\right)^{1/2}}$$

- Generalized eigenvalue problem:

$$\begin{pmatrix} 0 & C_{12} \\ C_{21} & 0 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \rho \begin{pmatrix} C_{11} & 0 \\ 0 & C_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}.$$

Canonical correlation analysis in feature space



- Given two random variables x_1 and x_2 and two RKHS $\mathcal{F}_1$ and $\mathcal{F}_2$, finds the pair of functions f_1, f_2 with maximum regularized correlation:

$$\max_{f_1, f_2 \in \mathcal{F}} \frac{\operatorname{cov}(f_1(X_1), f_2(X_2))}{(\operatorname{var}(f_1(X_1)) + \lambda_n \|f_1\|^2_{\mathcal{F}_1})^{1/2} (\operatorname{var}(f_2(X_2)) + \lambda_n \|f_2\|^2_{\mathcal{F}_2})^{1/2}}$$

- Criteria for independence (NB: independence $\neq$ uncorrelation)

Kernel Canonical Correlation Analysis

- Analogous derivation as Kernel PCA

- K_1, K_2 Gram matrices of $\{x_1^i\}$ and $\{x_2^i\}$

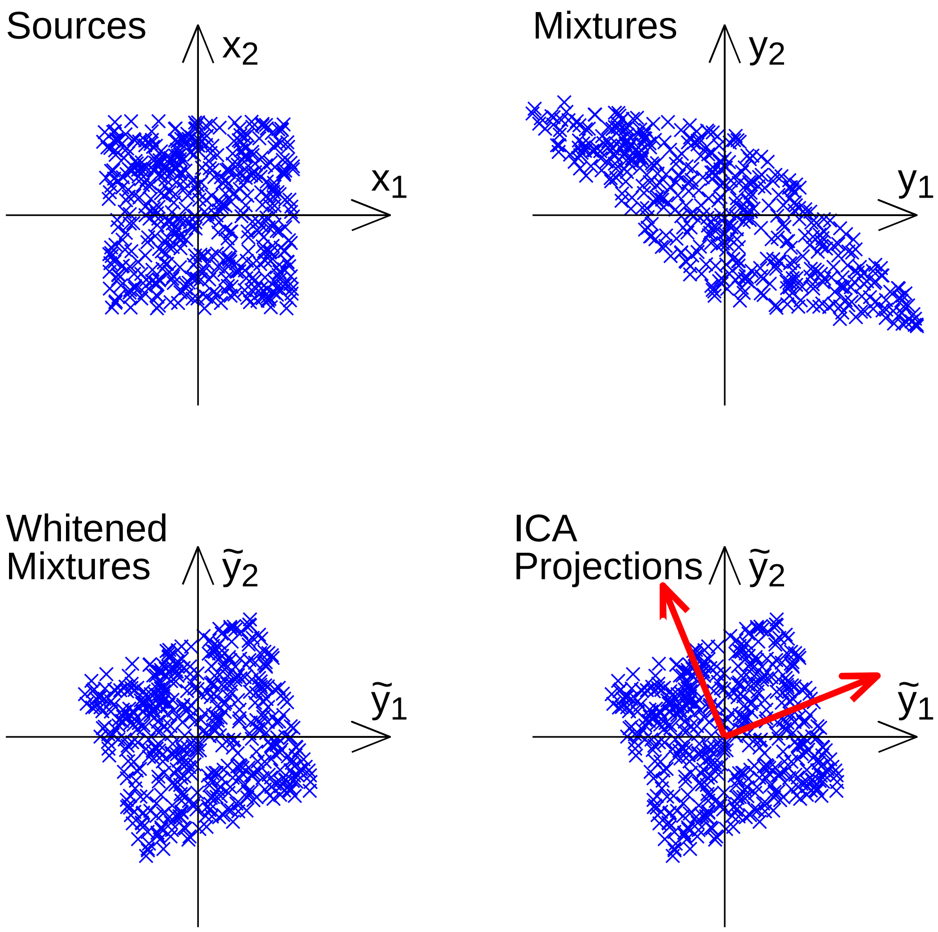
$$\max_{\alpha_1,\ \alpha_2 \in \Re^N} \frac{\alpha_1^T K_1 K_2 \alpha_2}{(\alpha_1^T (K_1^2 + \lambda K_1)\alpha_1)^{1/2} (\alpha_2^T (K_2^2 + \lambda K_2)\alpha_2)^{1/2}}$$

- Maximal generalized eigenvalue of

$$\begin{pmatrix} 0 & K_1 K_2 \\ K_2 K_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \rho \begin{pmatrix} K_1^2 + \lambda K_1 & 0 \\ 0 & K_2^2 + \lambda K_2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}$$

Kernel CCA
Application to ICA (Bach & Jordan, 2002)

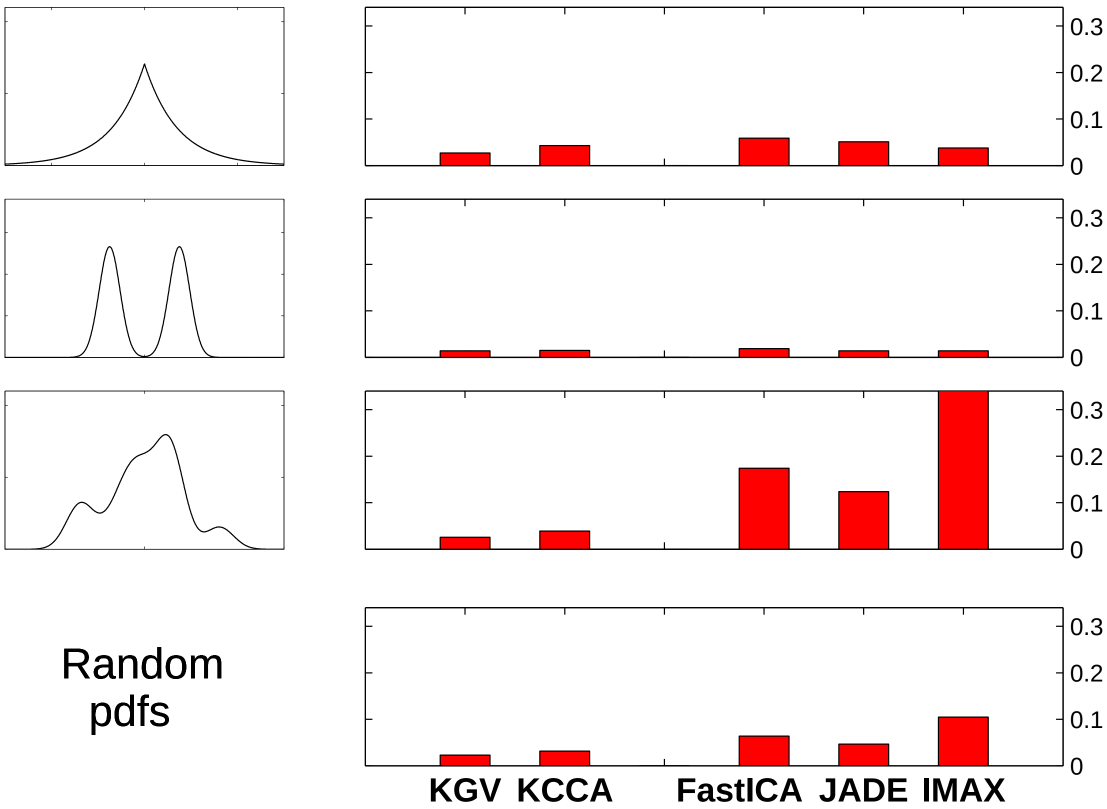
- Independent component analysis: linearly transform data such to get independent variables



Empirical results - Kernel ICA

- Comparison with other algorithms: FastICA (Hyvarinen,1999), Jade (Cardoso, 1998), Extended Infomax (Lee, 1999)

- Amari error : standard ICA distance from true sources



Course outline

1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Kernel design

- Principle: **kernel on $\mathcal{X}$ = space of functions on $\mathcal{X}$ + norm**

- Two main design principles

  1. Constructing kernels from kernels by algebraic operations
  2. Using usual algebraic/numerical tricks to perform efficient kernel computation with very high-dimensional feature spaces

- Operations: $k_1(x,y) = \langle \Phi_1(x), \Phi_1(y) \rangle$, $k_2(x,y) = \langle \Phi_2(x), \Phi_2(y) \rangle$

  – **Sum = concatenation of feature spaces**:

$$k_1(x,y) + k_2(x,y) = \left\langle \binom{\Phi_1(x)}{\Phi_2(x)}, \binom{\Phi_1(y)}{\Phi_2(y)} \right\rangle$$

  – **Product = tensor product of feature spaces**:

$$k_1(x,y)k_2(x,y) = \left\langle \Phi_1(x)\Phi_2(x)^\top, \Phi_1(y)\Phi_2(y)^\top \right\rangle$$

Classical kernels: kernels on vectors $x \in \mathbb{R}^d$

- Linear kernel $k(x, y) = x^\top y$
  – Linear functions
- Polynomial kernel $k(x, y) = (1 + x^\top y)^d$
  – Polynomial functions
- Gaussian kernel $k(x, y) = \exp(-\alpha \|x - y\|^2)$
  – Smooth functions
- Data are not always vectors!

Efficient ways of computing large sums

- Goal: $\Phi(x) \in \mathbb{R}^p$ high-dimensional, compute $\sum_{i=1}^{p} \Phi_i(x)\Phi_i(y)$ **in** $o(p)$

- **Sparsity**: many $\Phi_i(x)$ equal to zero (example: pyramid match kernel)

- **Factorization and recursivity**: replace sums of many products by product of few sums (example: polynomial kernel, graph kernel)

$$(1 + x^\top y)^d = \sum_{\alpha_1+\cdots+\alpha_k \leqslant d} \binom{d}{\alpha_1, \ldots, \alpha_k} (x_1 y_1)^{\alpha_1} \cdots (x_k y_k)^{\alpha_k}$$

Kernels over (labelled) sets of points

- Common situation in computer vision (e.g., interest points)
- Simple approach: compute averages/histograms of certain features
  - valid kernels over histograms h and h' (Hein and Bousquet, 2004)
  - **intersection**: $\sum_i \min(h_i, h'_i)$, **chi-square**: $\exp\left(-\alpha \sum_i \frac{(h_i - h'_i)^2}{h_i + h'_i}\right)$

Kernels over (labelled) sets of points

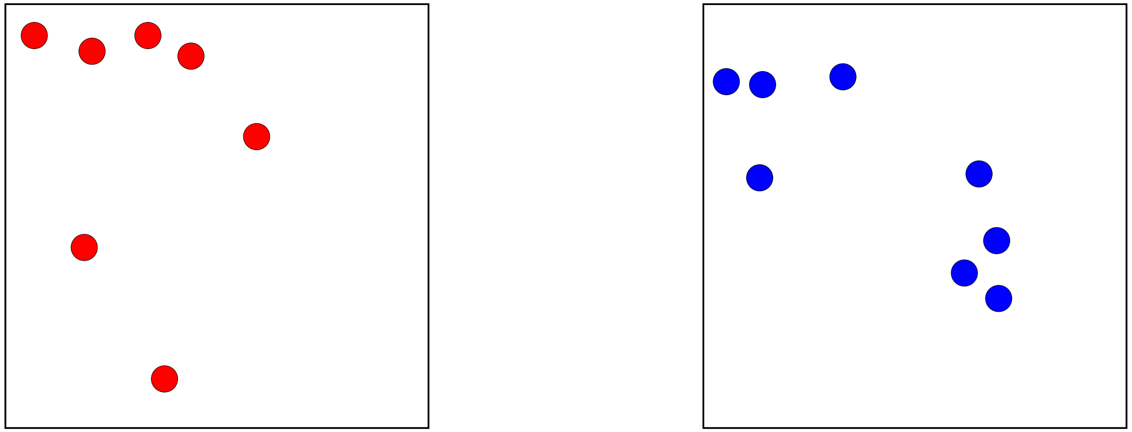
- Common situation in computer vision (e.g., interest points)

- Simple approach: compute averages/histograms of certain features
  - valid kernels over histograms h and h' (Hein and Bousquet, 2004)
  - **intersection**: $\sum_i \min(h_i, h_i')$, **chi-square**: $\exp\left(-\alpha \sum_i \frac{(h_i - h_i')^2}{h_i + h_i'}\right)$

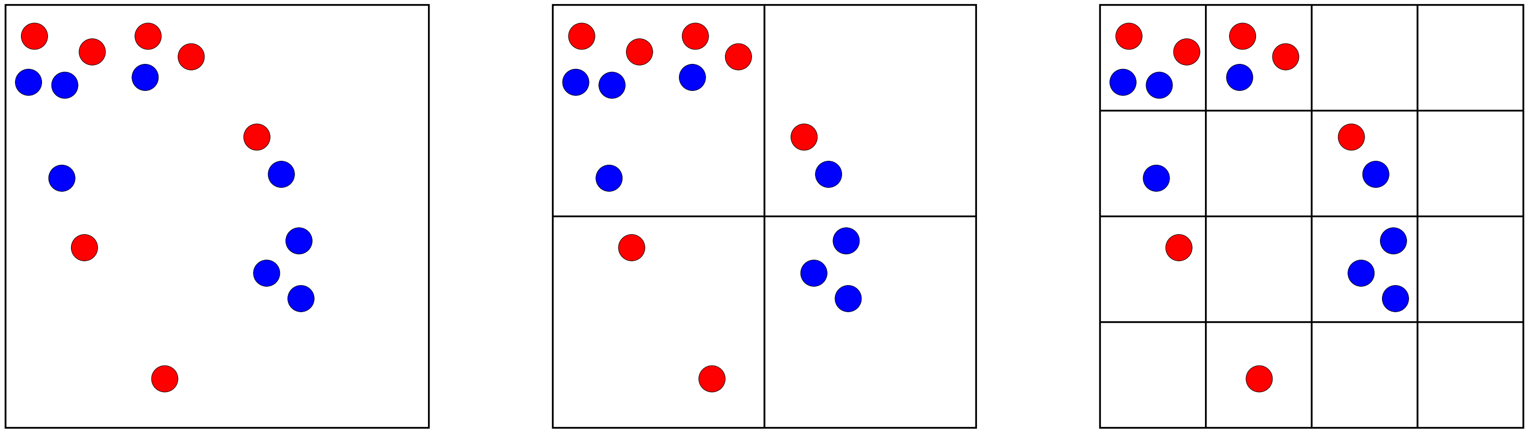
- Pyramid match (Grauman and Darrell, 2007): efficiently introducing localization
  - Form a regular pyramid on top of the image
  - Count the number of common elements in each bin
  - Give a weight to each bin
  - Many bins but most of them are empty
    $\Rightarrow$ use sparsity to compute kernel efficiently

Pyramid match kernel
(Grauman and Darrell, 2007; Lazebnik et al., 2006)

- Two sets of points



- Counting matches at several scales: 7, 5, 4



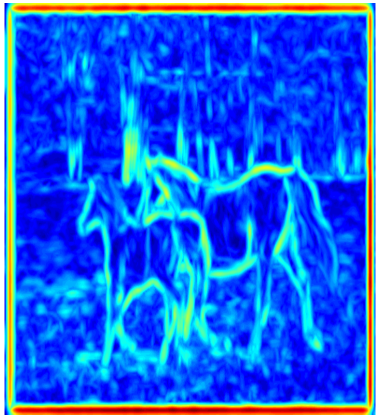
Kernels from segmentation graphs

- Goal of segmentation: extract objects of interest
- Many methods available, ...
  - ... but, rarely find the object of interest entirely
- Segmentation graphs
  - Allows to work on "more reliable" over-segmentation
  - Going to a large square grid (millions of pixels) to a small graph (dozens or hundreds of regions)
- How to build a kernel over segmenation graphs?
  - NB: more generally, kernelizing existing representations?

Segmentation by watershed transform (Meyer, 2001)

image

gradient



watershed

287 segments

64 segments

10 segments

Segmentation by watershed transform (Meyer, 2001)

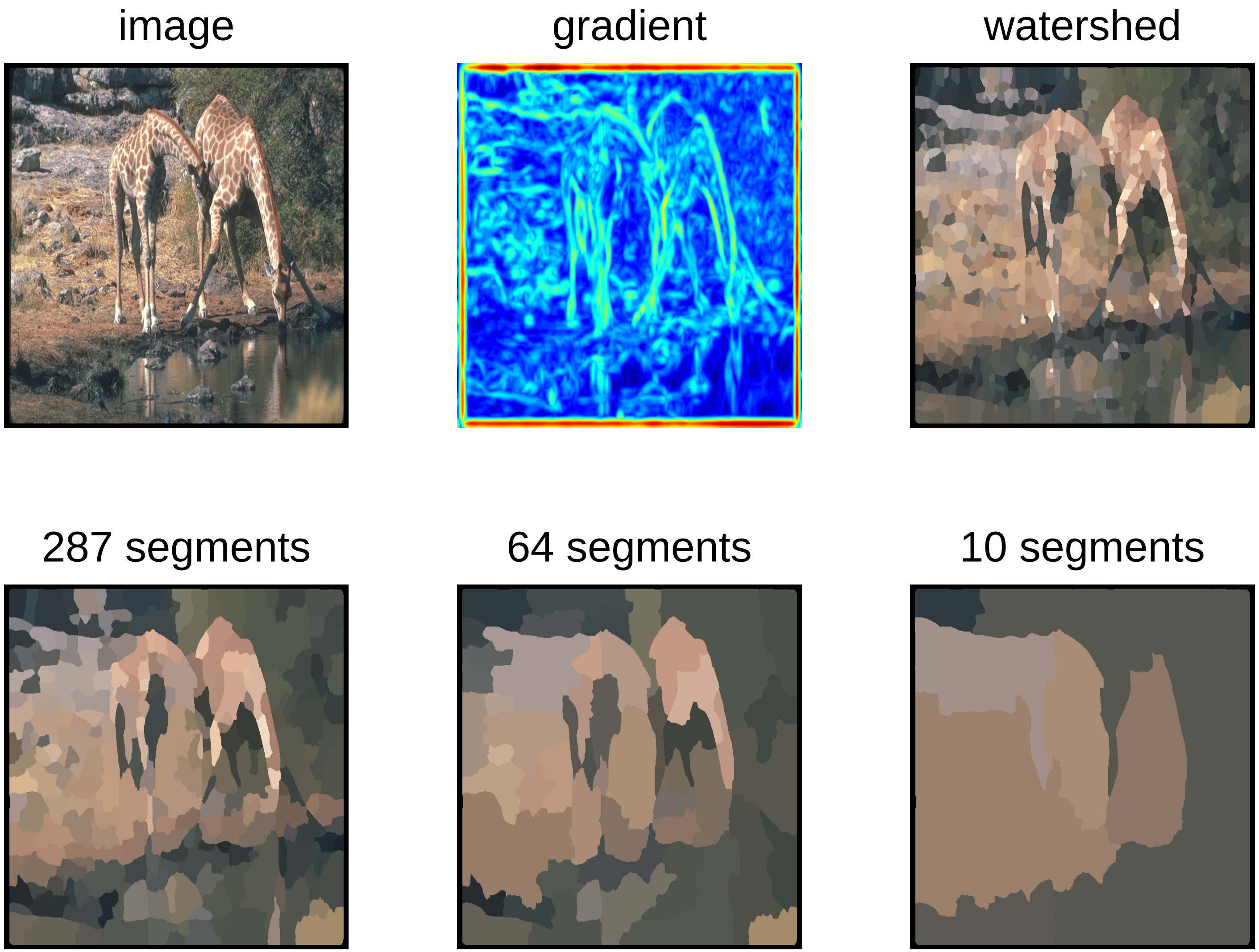


Image as a segmentation graph

- Labelled undirected graph
  - Vertices: connected segmented regions
  - Edges: between spatially neighboring regions
  - Labels: region pixels

$\Rightarrow$

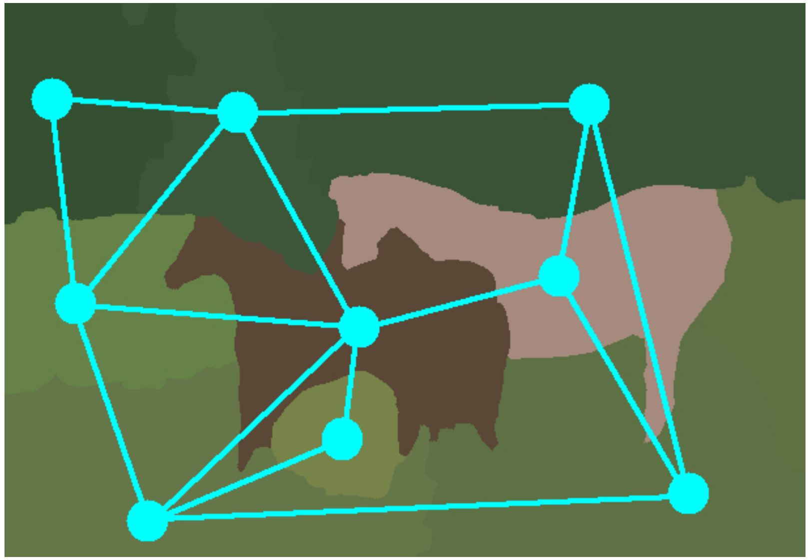


Image as a segmentation graph

- Labelled undirected graph
  - Vertices: connected segmented regions
  - Edges: between spatially neighboring regions
  - Labels: region pixels

- Difficulties
  - Extremely high-dimensional labels
  - Planar undirected graph
  - Inexact matching

- **Graph kernels** (Gärtner et al., 2003; Kashima et al., 2004; Harchaoui and Bach, 2007) provide an elegant and efficient solution

Kernels between structured objects

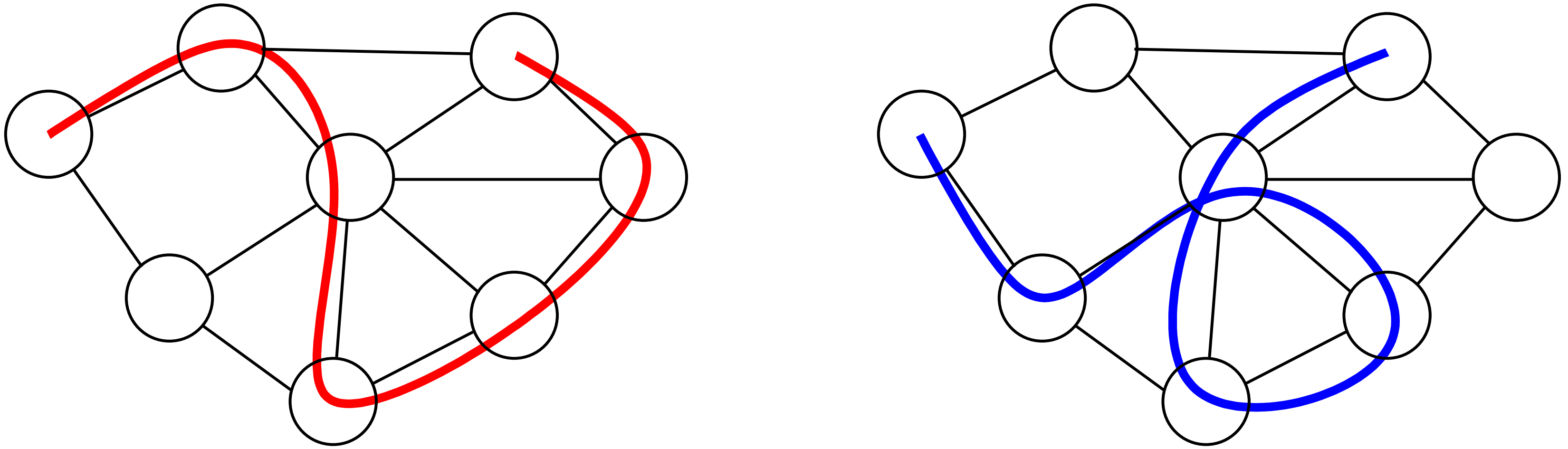
Strings, graphs, etc... (Shawe-Taylor and Cristianini, 2004)

- Numerous applications (text, bio-informatics, speech, vision)
- Common design principle: **enumeration of subparts** (Haussler, 1999; Watkins, 1999)
  - Efficient for strings
  - Possibility of gaps, partial matches, very efficient algorithms
- Most approaches fails for general graphs (even for undirected trees!)
  - NP-Hardness results (Ramon and Gärtner, 2003)
  - Need specific set of subparts

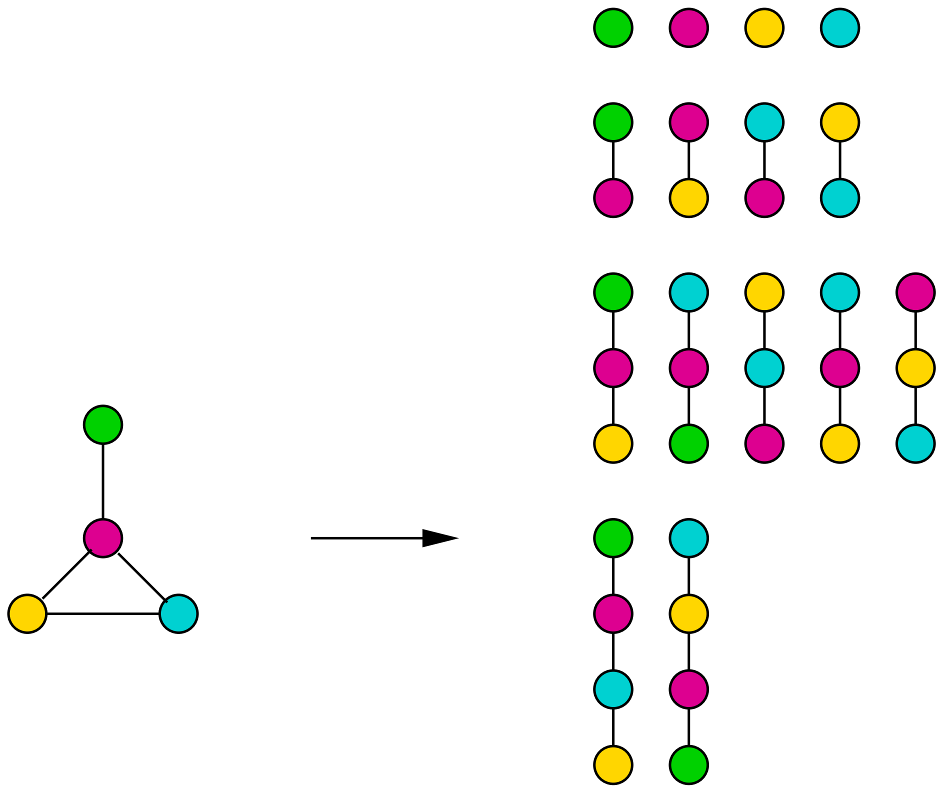
Paths and walks

- Given a graph G,
  - A path is a sequence of distinct neighboring vertices
  - A walk is a sequence of neighboring vertices

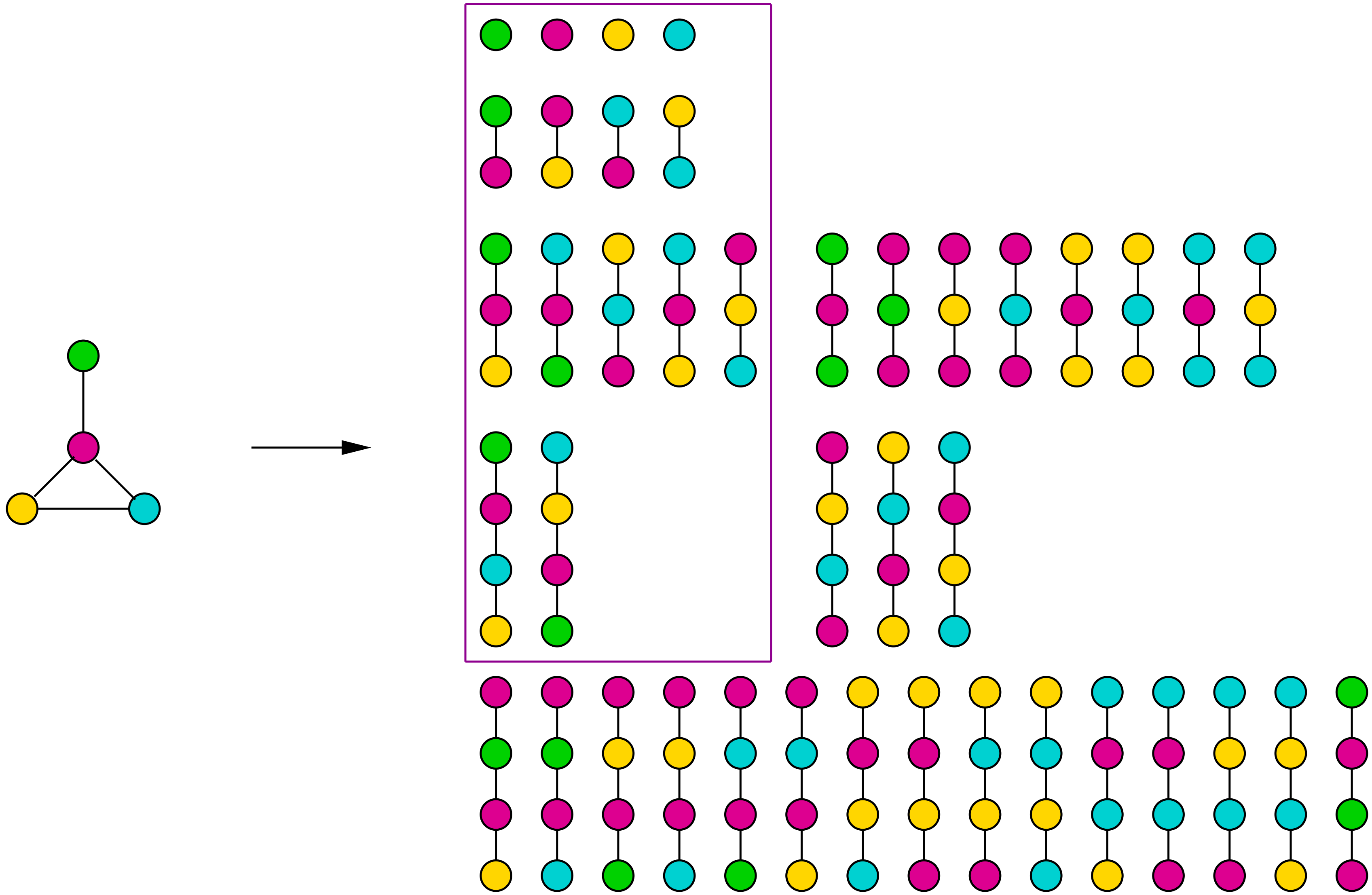
- Apparently similar notions



Paths



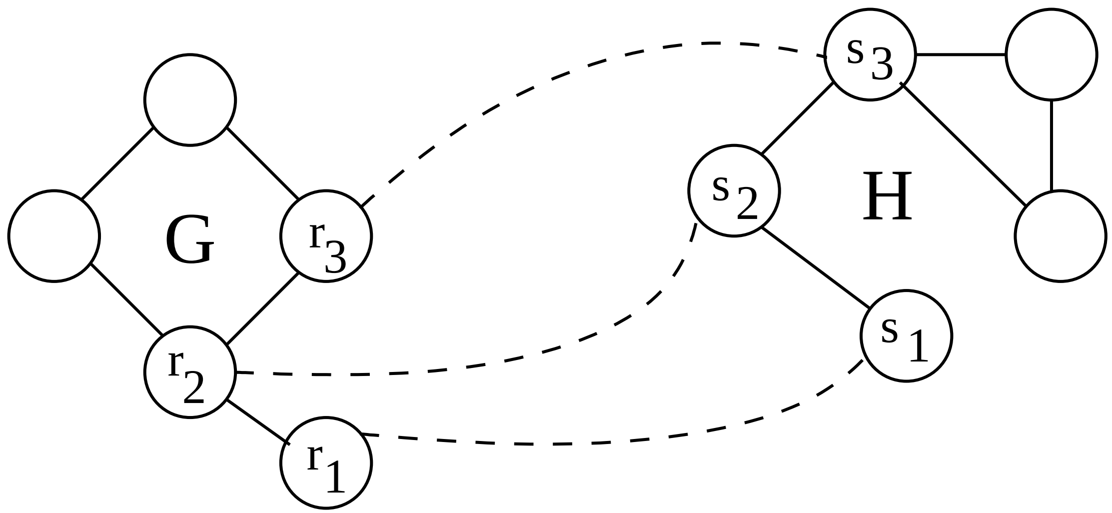
Walks



Walk kernel (Kashima et al., 2004; Borgwardt et al., 2005)

- $\mathcal{W}^p_{\mathbf{G}}$ (resp. $\mathcal{W}^p_{\mathbf{H}}$) denotes the set of walks of length p in $\mathbf{G}$ (resp. $\mathbf{H}$)
- Given *basis kernel* on labels $k(\ell, \ell')$
- p-th order walk kernel:

$$k^p_{\mathcal{W}}(\mathbf{G}, \mathbf{H}) = \sum_{\substack{(r_1, \ldots, r_p) \in \mathcal{W}^p_{\mathbf{G}} \\ (s_1, \ldots, s_p) \in \mathcal{W}^p_{\mathbf{H}}}} \prod_{i=1}^{p} k(\ell_{\mathbf{G}}(r_i), \ell_{\mathbf{H}}(s_i)).$$



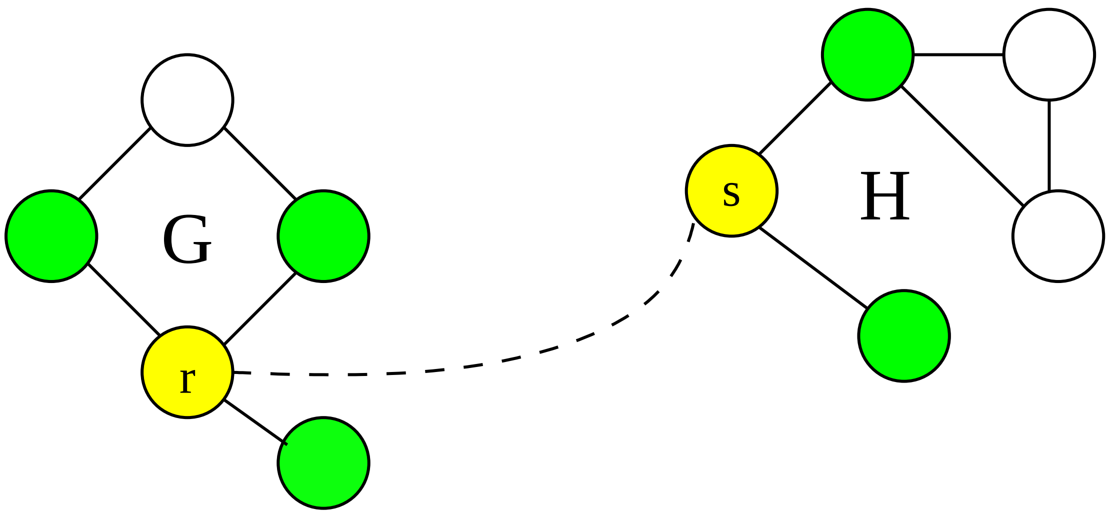
Dynamic programming for the walk kernel

- Dynamic programming in $O(pd_{\mathbf{G}}d_{\mathbf{H}}n_{\mathbf{G}}n_{\mathbf{H}})$

- $k^p_{\mathcal{W}}(\mathbf{G},\mathbf{H},r,s)$ = sum restricted to walks starting at r and s

- recursion between $p-1$-th walk and p-th walk kernel

$$k^p_{\mathcal{W}}(\mathbf{G},\mathbf{H},r,s)=k(\ell_{\mathbf{G}}(r),\ell_{\mathbf{H}}(s))\sum_{\substack{r'\in\mathcal{N}_{\mathbf{G}}(r)\\ s'\in\mathcal{N}_{\mathbf{H}}(s)}} k^{p-1}_{\mathcal{W}}(\mathbf{G},\mathbf{H},r',s').$$



Dynamic programming for the walk kernel

- Dynamic programming in $O(pd_{\mathbf{G}}d_{\mathbf{H}}n_{\mathbf{G}}n_{\mathbf{H}})$

- $k_{\mathcal{W}}^{p}(\mathbf{G},\mathbf{H},r,s)$ = sum restricted to walks starting at r and s

- recursion between $p-1$-th walk and p-th walk kernel

$$k_{\mathcal{W}}^{p}(\mathbf{G},\mathbf{H},r,s) = k(\ell_{\mathbf{G}}(r),\ell_{\mathbf{H}}(s)) \sum_{\substack{r' \in \mathcal{N}_{\mathbf{G}}(r) \\ s' \in \mathcal{N}_{\mathbf{H}}(s)}} k_{\mathcal{W}}^{p-1}(\mathbf{G},\mathbf{H},r',s')$$

- Kernel obtained as $k_{\mathcal{T}}^{p,\alpha}(\mathbf{G},\mathbf{H}) = \sum_{r \in \mathcal{V}_{\mathbf{G}}, s \in \mathcal{V}_{\mathbf{H}}} k_{\mathcal{T}}^{p,\alpha}(\mathbf{G},\mathbf{H},r,s)$

Extensions of graph kernels

- Main principle: **compare all possible subparts of the graphs**
- Going from paths to subtrees
  - Extension of the concept of walks $\Rightarrow$ tree-walks (Ramon and Gärtner, 2003)
- Similar dynamic programming recursions (Harchaoui and Bach, 2007)
- Need to play around with subparts to obtain efficient recursions
  - Do we actually need positive definiteness?

Performance on Corel14 (Harchaoui and Bach, 2007)

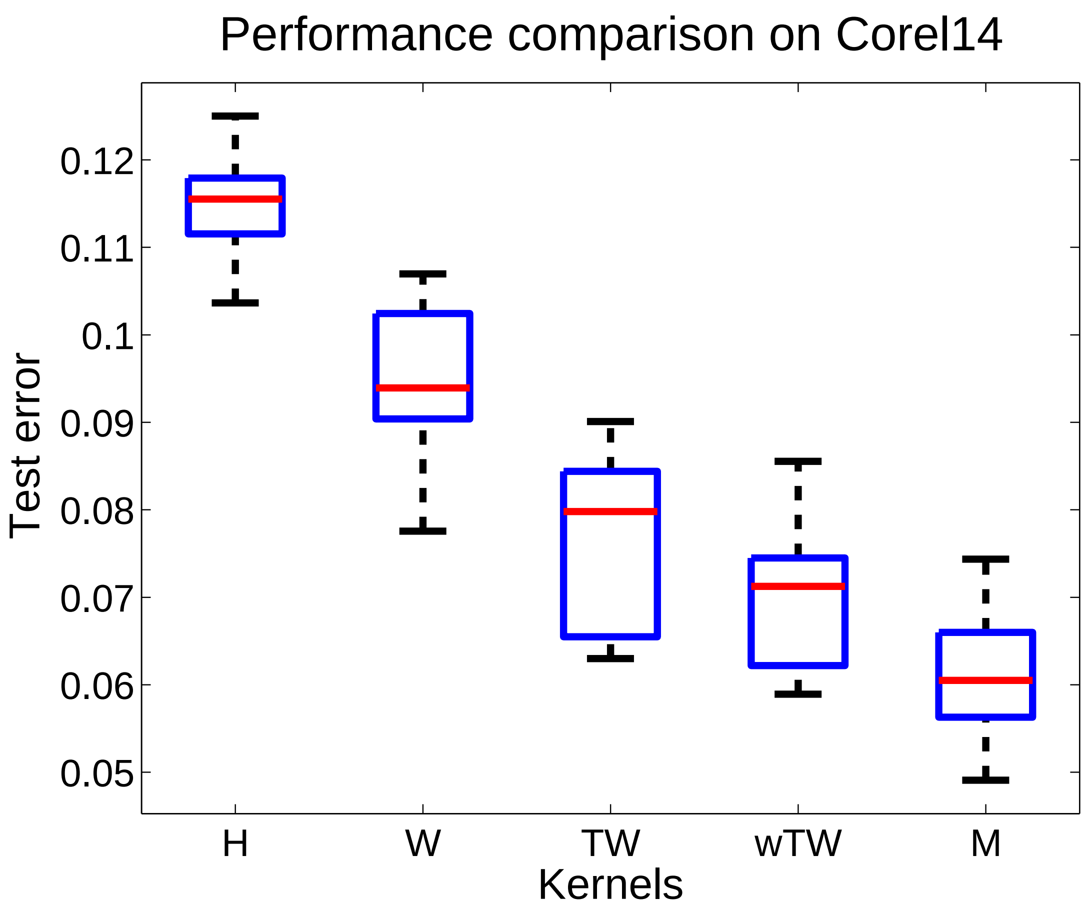
- `Corel14`: 1400 natural images with 14 classes

Performance on Corel14 (Harchaoui & Bach, 2007)
Error rates

- Histogram kernels (**H**)
- Walk kernels (**W**)
- Tree-walk kernels (**TW**)
- Weighted tree-walks (**wTW**)
- MKL (**M**)



Kernel methods - Summary

- Kernels and representer theorems
  - Clear distinction between representation/algorithms
- Algorithms
  - Two formulations (primal/dual)
  - Logistic or SVM?
- Kernel design
  - Very large feature spaces with efficient kernel evaluations

Course outline

1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Supervised learning and regularization

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$
- Minimize with respect to function $f : \mathcal{X} \to \mathcal{Y}$:

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

Error on data + Regularization

Loss & function space ? Norm ?

- Two theoretical/algorithmic issues:

1. Loss
2. **Function space / norm**

Regularizations

- **Main goal: avoid overfitting**

- **Two main lines of work**:

  1. Euclidean and Hilbertian norms (i.e., ℓ_2-norms)
     - Possibility of non linear predictors
     - Non parametric supervised learning and kernel methods
     - Well developped theory and algorithms (see, e.g., Wahba, 1990; Schölkopf and Smola, 2001; Shawe-Taylor and Cristianini, 2004)

Regularizations

- **Main goal: avoid overfitting**

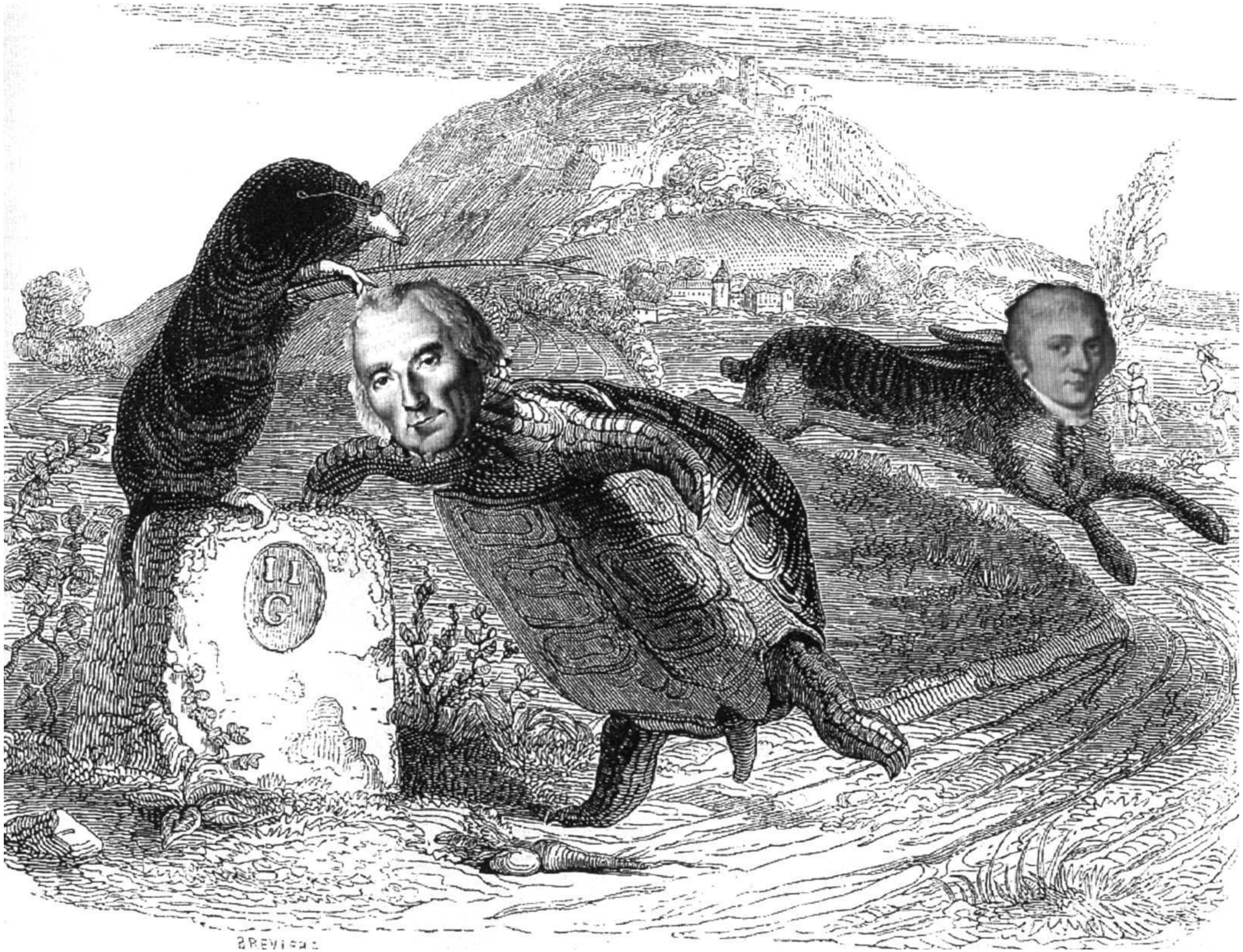
- **Two main lines of work**:

  1. Euclidean and Hilbertian norms (i.e., ℓ_2-norms)
     - Possibility of non linear predictors
     - Non parametric supervised learning and kernel methods
     - Well developped theory and algorithms (see, e.g., Wahba, 1990; Schölkopf and Smola, 2001; Shawe-Taylor and Cristianini, 2004)
  2. Sparsity-inducing norms
     - Usually restricted to linear predictors on vectors $f(x) = w^\top x$
     - Main example: ℓ_1-norm $\|w\|_1 = \sum_{i=1}^p |w_i|$
     - Perform model selection as well as regularization
     - **Theory and algorithms "in the making"**

ℓ_2-norm vs. ℓ_1-norm

- ℓ_1-norms lead to interpretable models
- ℓ_2-norms can be run implicitly with very large feature spaces
- **Algorithms**:
  - Smooth convex optimization vs. nonsmooth convex optimization
- **Theory**:
  - better predictive performance?

ℓ_2 vs. ℓ_1 - Gaussian hare vs. Laplacian tortoise



- First-order methods (Fu, 1998; Wu and Lange, 2008)
- Homotopy methods (Markowitz, 1956; Efron et al., 2004)

Lasso - Two main recent theoretical results

1. **Support recovery condition** (Zhao and Yu, 2006; Wainwright, 2006; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if

$$\|\mathbf{Q}_{\mathbf{J}^c\mathbf{J}}\mathbf{Q}_{\mathbf{J}\mathbf{J}}^{-1}\text{sign}(\mathbf{w}_\mathbf{J})\|_\infty \leqslant 1,$$

where $\mathbf{Q} = \lim_{n\to+\infty}\frac{1}{n}\sum_{i=1}^n x_i x_i^\top \in \mathbb{R}^{p\times p}$ and $\mathbf{J} = \text{Supp}(\mathbf{w})$

Lasso - Two main recent theoretical results

1. **Support recovery condition** (Zhao and Yu, 2006; Wainwright, 2006; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if
$$\|\mathbf{Q}_{\mathbf{J}^c\mathbf{J}}\mathbf{Q}_{\mathbf{J}\mathbf{J}}^{-1}\mathrm{sign}(\mathbf{w}_\mathbf{J})\|_\infty \leqslant 1,$$
where $\mathbf{Q} = \lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^n x_i x_i^\top \in \mathbb{R}^{p\times p}$ and $\mathbf{J} = \mathrm{Supp}(\mathbf{w})$

2. **Exponentially many irrelevant variables** (Zhao and Yu, 2006; Wainwright, 2006; Bickel et al., 2009; Lounici, 2008; Meinshausen and Yu, 2008): under appropriate assumptions, consistency is possible as long as
$$\log p = O(n)$$

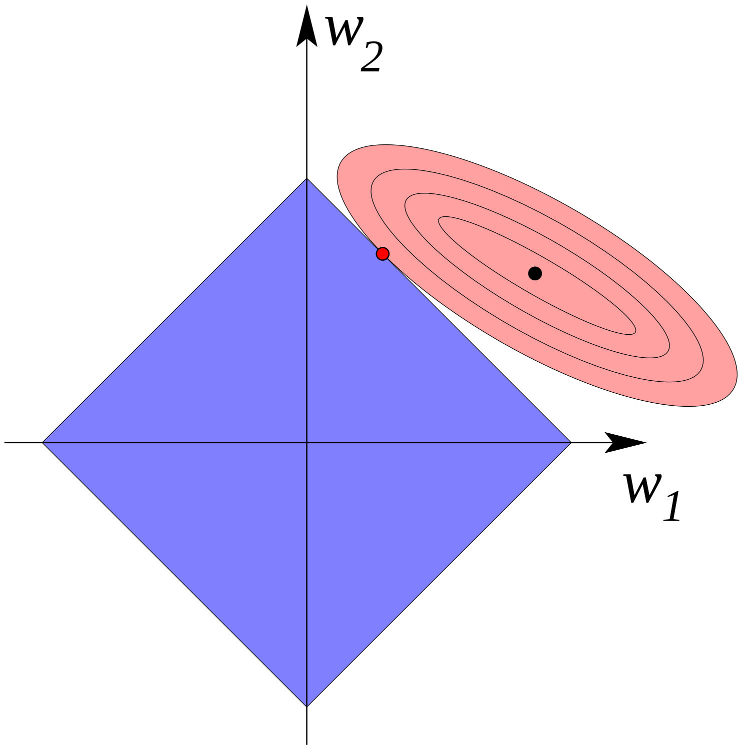
Going beyond the Lasso

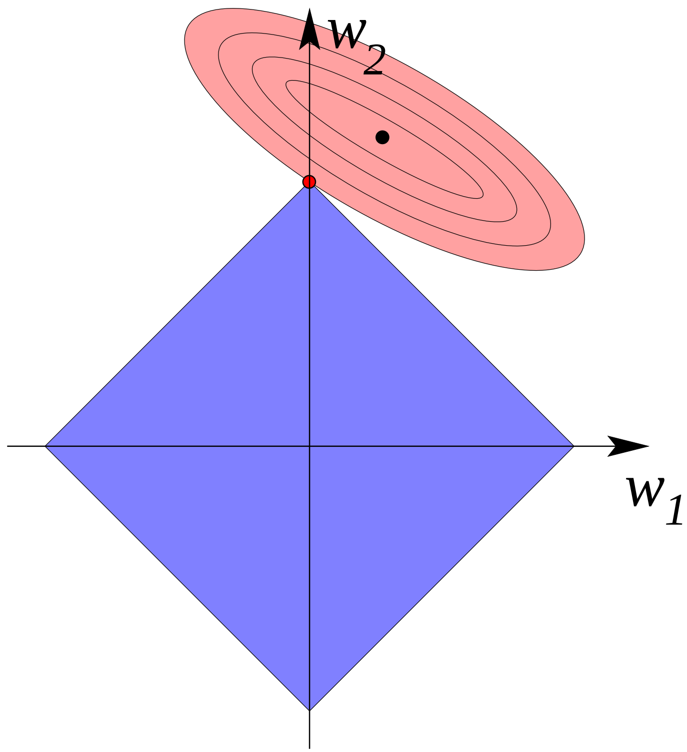
- ℓ_1-norm for **linear** feature selection in **high dimensions**
  - Lasso usually not applicable directly
- **Non-linearities**
- **Dealing with exponentially many features**
- **Sparse learning on matrices**

Why ℓ_1-norm constraints leads to sparsity?

- Example: minimize quadratic function $Q(w)$ subject to $\|w\|_1 \leqslant T$.
  - coupled soft thresholding

- Geometric interpretation
  - NB : penalizing is "equivalent" to constraining





ℓ_1-norm regularization (linear setting)

- Data: covariates $x_i \in \mathbb{R}^p$, responses $y_i \in \mathcal{Y}$, $i = 1, \ldots, n$

- Minimize with respect to loadings/weights $w \in \mathbb{R}^p$:

$$J(w) = \underbrace{\sum_{i=1}^{n} \ell(y_i, w^\top x_i)}_{\text{Error on data}} \quad + \quad \underbrace{\lambda \|w\|_1}_{\text{Regularization}}$$

- Including a constant term b? Penalizing or constraining?

- square loss $\Rightarrow$ basis pursuit in signal processing (Chen et al., 2001), Lasso in statistics/machine learning (Tibshirani, 1996)

First order methods for convex optimization on $\mathbb{R}^p$

Smooth optimization

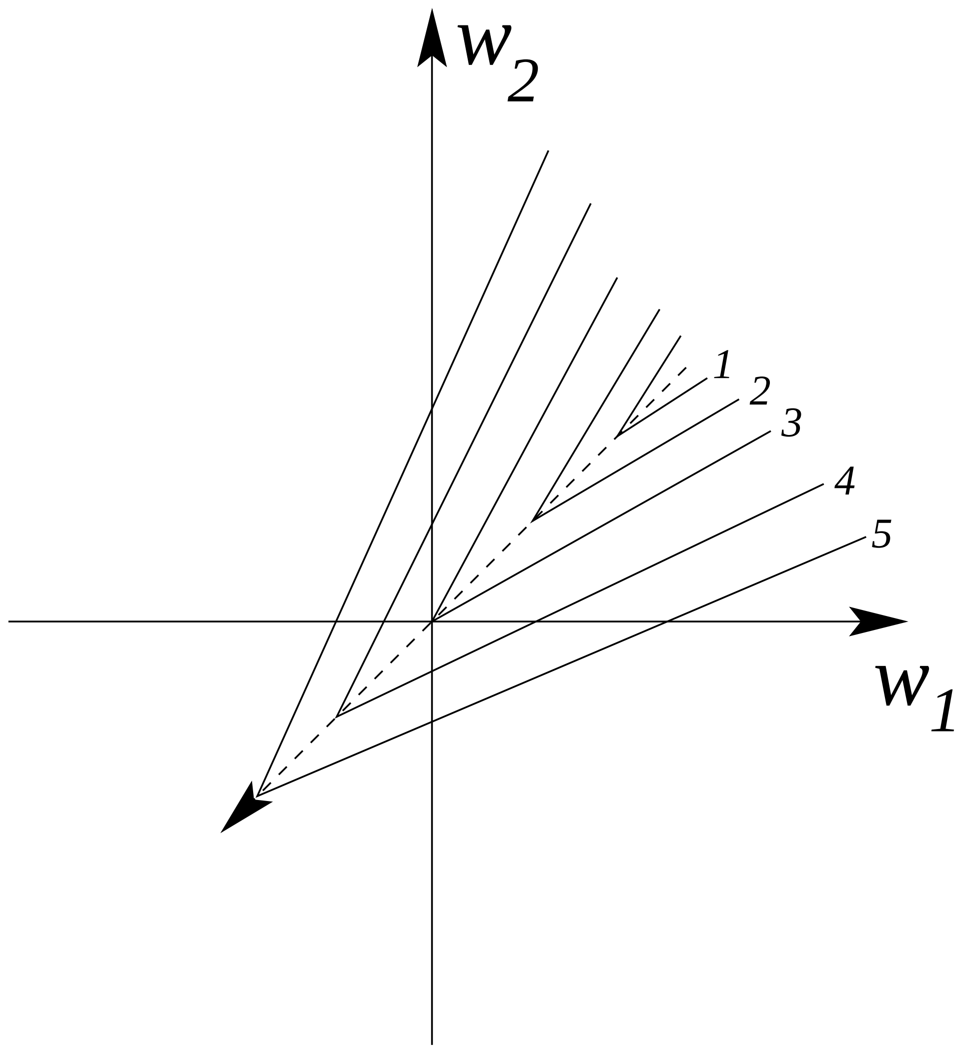
- **Gradient descent**: $w_{t+1} = w_t - \alpha_t \nabla J(w_t)$
  - with line search: search for a decent (not necessarily best) α_t
  - fixed diminishing step size, e.g., $\alpha_t = a(t+b)^{-1}$
- Convergence of $f(w_t)$ to $f^* = \min_{w \in \mathbb{R}^p} f(w)$ (Nesterov, 2003)
  - f convex and M-Lipschitz: $f(w_t) - f^* = O\big(M/\sqrt{t}\big)$
  - and, differentiable with L-Lipschitz gradient: $f(w_t) - f^* = O\big(L/t\big)$
  - and, f μ-strongly convex: $f(w_t) - f^* = O\big(L \exp(-4t\frac{\mu}{L})\big)$
- $\frac{\mu}{L}$ = condition number of the optimization problem
- Coordinate descent: similar properties
- NB: "optimal scheme" $f(w_t) - f^* = O\big(L \min\{\exp(-4t\sqrt{\mu/L}), t^{-2}\}\big)$

First-order methods for convex optimization on $\mathbb{R}^p$
Non smooth optimization

- First-order methods for non differentiable objective
  - Subgradient descent: $w_{t+1} = w_t - \alpha_t g_t$, with $g_t \in \partial J(w_t)$, i.e., such that $\forall \Delta, g_t^\top \Delta \leqslant \nabla J(w_t, \Delta)$
    - with exact line search: not always convergent (see counter-example)
    - diminishing step size, e.g., $\alpha_t = a(t+b)^{-1}$: convergent
  - Coordinate descent: not always convergent (show counter-example)

- Convergence rates (f convex and M-Lipschitz): $f(w_t) - f^* = O\big(\frac{M}{\sqrt{t}}\big)$

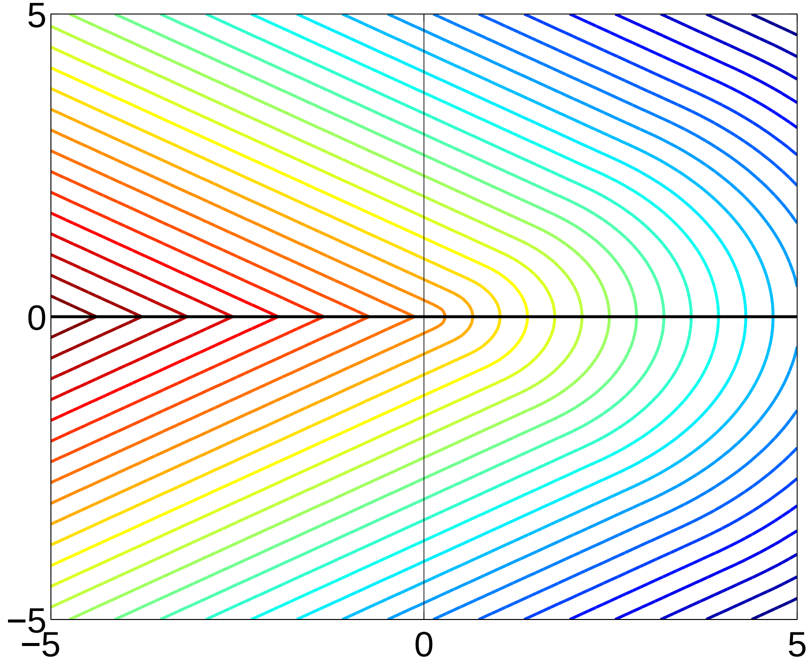
Counter-example
Coordinate descent for nonsmooth objectives



Counter-example (Bertsekas, 1995)
Steepest descent for nonsmooth objectives

- $q(x_1, x_2) = \begin{cases} -5(9x_1^2 + 16x_2^2)^{1/2} & \text{if } x_1 > |x_2| \\ -(9x_1 + 16|x_2|)^{1/2} & \text{if } x_1 \leqslant |x_2| \end{cases}$

- Steepest descent starting from any x such that $x_1 > |x_2| > (9/16)^2 |x_1|$



Regularized problems - Proximal methods

- Gradient descent as a proximal method (differentiable functions)
  - $w_{t+1} = \arg\min_{w \in \mathbb{R}^p} J(w_t) + (w - w_t)^\top \nabla J(w_t) + \frac{L}{2}\|w - w_t\|_2^2$
  - $w_{t+1} = w_t - \frac{1}{L}\nabla J(w_t)$

- Problems of the form: $\boxed{\min_{w \in \mathbb{R}^p} L(w) + \lambda\Omega(w)}$
  - $w_{t+1} = \arg\min_{w \in \mathbb{R}^p} L(w_t) + (w - w_t)^\top \nabla L(w_t) + \lambda\Omega(w) + \frac{L}{2}\|w - w_t\|_2^2$
  - Thresholded gradient descent

- Similar convergence rates than smooth optimization
  - Acceleration methods (Nesterov, 2007; Beck and Teboulle, 2009)
  - **depends on the condition number of the loss**

Second order methods

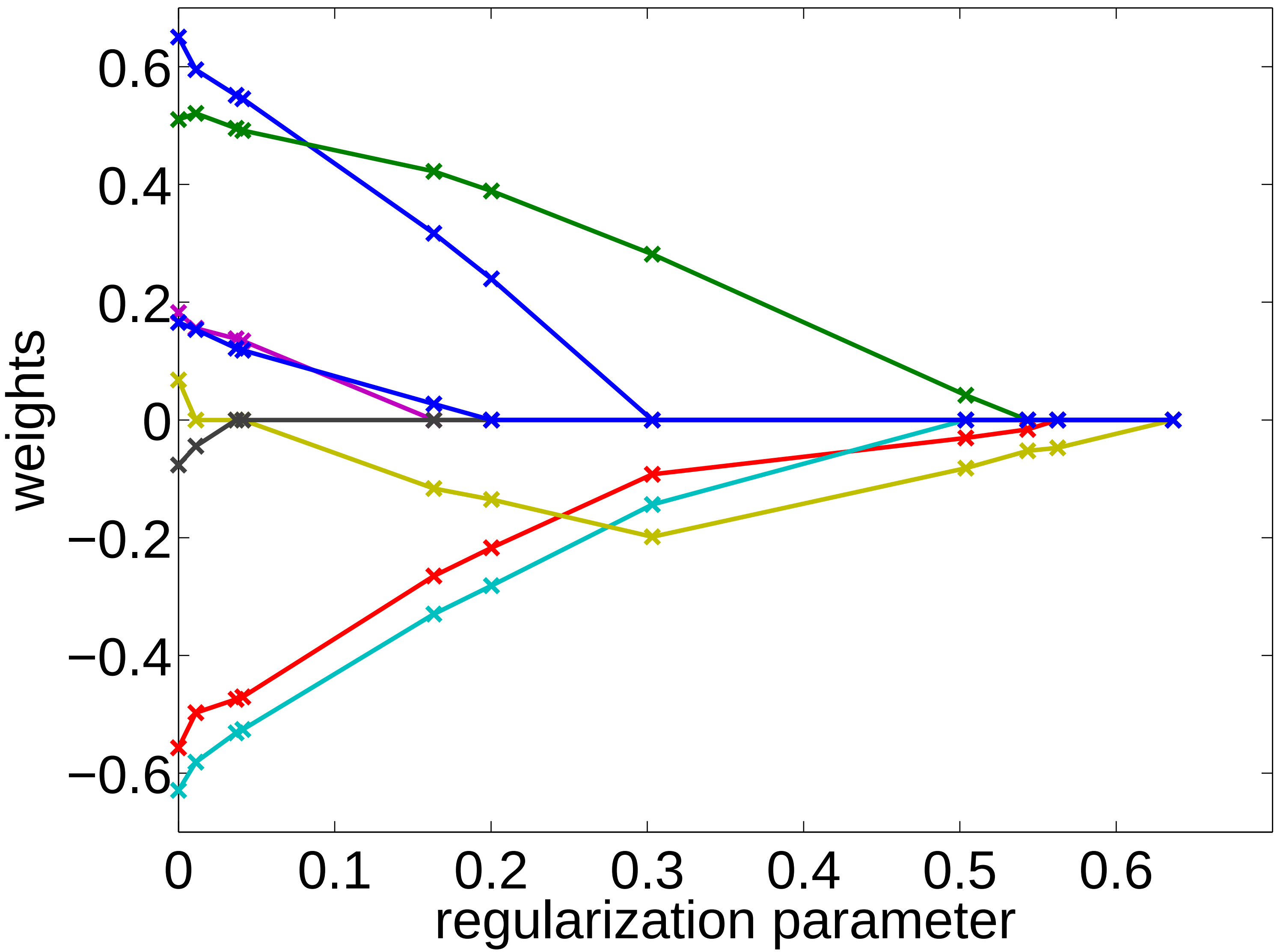
- Differentiable case
  - Newton: $w_{t+1} = w_t - \alpha_t H_t^{-1} g_t$
    * Traditional: $\alpha_t = 1$, but non globally convergent
    * globally convergent with line search for α_t (see Boyd, 2003)
    * $O(\log\log(1/\varepsilon))$ (slower) iterations
  - Quasi-newton methods (see Bonnans et al., 2003)

- Non differentiable case (interior point methods)
  - Smoothing of problem + second order methods
    * See example later and (Boyd, 2003)
    * Theoretically $O(\sqrt{p})$ Newton steps, usually $O(1)$ Newton steps

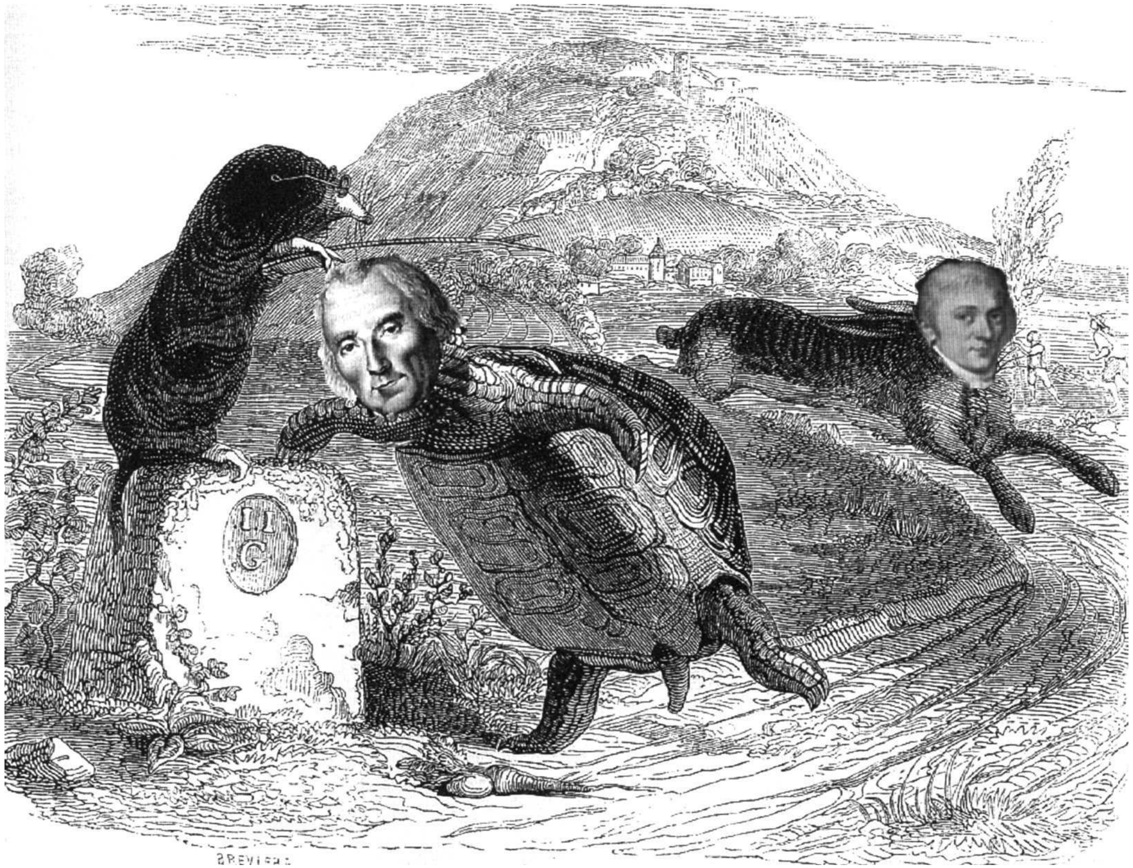
First order or second order methods for machine learning?

- objecive defined as average (i.e., up to $n^{-1/2}$): no need to optimize up to 10^{-16}!
  - Second-order: slower but worryless
  - First-order: faster but care must be taken regarding convergence
- Rule of thumb
  - Small scale $\Rightarrow$ second order
  - Large scale $\Rightarrow$ first order
  - Unless dedicated algorithm using structure (like for the Lasso)
- See Bottou and Bousquet (2008) for further details

Piecewise linear paths



Algorithms for ℓ_1-norms (square loss): Gaussian hare vs. Laplacian tortoise



- Coordinate descent: $O(pn)$ per iterations for ℓ_1 and ℓ_2
- "Exact" algorithms: $O(kpn)$ for ℓ_1 **vs.** $O(p^2n)$ for ℓ_2

Additional methods - Softwares

- Many contributions in signal processing, optimization, machine learning
  - Extensions to stochastic setting (Bottou and Bousquet, 2008)
- Extensions to other sparsity-inducing norms
  - Computing proximal operator
- **Softwares**
  - Many available codes
  - SPAMS (SPArse Modeling Software)
    `http://www.di.ens.fr/willow/SPAMS/`

Course outline

1. **Losses for particular machine learning tasks**
    - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
    - Kernels and representer theorem
    - Convex duality, optimization and algorithms
    - Kernel methods
    - Kernel design
3. **Regularization by sparsity-inducing norms**
    - ℓ_1-norm regularization
    - Multiple kernel learning
    - Theoretical results
    - Learning on matrices

Theoretical results - Square loss

- Main assumption: data generated from a certain sparse $\mathbf{w}$

- Three main problems:

  1. **Regular consistency**: convergence of estimator $\hat{w}$ to $\mathbf{w}$, i.e., $\|\hat{w} - \mathbf{w}\|$ tends to zero when n tends to ∞
  2. **Model selection consistency**: convergence of the sparsity pattern of $\hat{w}$ to the pattern $\mathbf{w}$
  3. **Efficiency**: convergence of predictions with $\hat{w}$ to the predictions with $\mathbf{w}$, i.e., $\frac{1}{n}\|X\hat{w} - X\mathbf{w}\|_2^2$ tends to zero

- Main results:

  – **Condition for model consistency (support recovery)**
  – **High-dimensional inference**

Model selection consistency (Lasso)

- Assume $\mathbf{w}$ sparse and denote $\mathbf{J} = \{j, \mathbf{w}_j \neq 0\}$ the nonzero pattern
- **Support recovery condition** (Zhao and Yu, 2006; Wainwright, 2006; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if

$$\boxed{\|\mathbf{Q}_{\mathbf{J}^c\mathbf{J}}\mathbf{Q}_{\mathbf{J}\mathbf{J}}^{-1}\mathrm{sign}(\mathbf{w}_\mathbf{J})\|_\infty \leqslant 1}$$

where $\mathbf{Q} = \lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^n x_i x_i^\top \in \mathbb{R}^{p\times p}$ and $\mathbf{J} = \mathrm{Supp}(\mathbf{w})$

Model selection consistency (Lasso)

- Assume $\mathbf{w}$ sparse and denote $\mathbf{J} = \{j, \mathbf{w}_j \neq 0\}$ the nonzero pattern
- **Support recovery condition** (Zhao and Yu, 2006; Wainwright, 2006; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if
$$\boxed{\|\mathbf{Q}_{\mathbf{J}^c\mathbf{J}}\mathbf{Q}_{\mathbf{J}\mathbf{J}}^{-1}\mathrm{sign}(\mathbf{w}_\mathbf{J})\|_\infty \leqslant 1}$$
where $\mathbf{Q} = \lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^n x_i x_i^\top \in \mathbb{R}^{p\times p}$ and $\mathbf{J} = \mathrm{Supp}(\mathbf{w})$
- Condition depends on $\mathbf{w}$ and $\mathbf{J}$ (may be relaxed)
  - may be relaxed by maximizing out $\mathrm{sign}(\mathbf{w})$ or $\mathbf{J}$
- Valid in low and high-dimensional settings
- Requires lower-bound on magnitude of nonzero $\mathbf{w}_j$

Model selection consistency (Lasso)

- Assume $\mathbf{w}$ sparse and denote $\mathbf{J} = \{j, \mathbf{w}_j \neq 0\}$ the nonzero pattern

- **Support recovery condition** (Zhao and Yu, 2006; Wainwright, 2006; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if

$$\boxed{\|\mathbf{Q}_{\mathbf{J}^c\mathbf{J}}\mathbf{Q}_{\mathbf{J}\mathbf{J}}^{-1}\operatorname{sign}(\mathbf{w}_\mathbf{J})\|_\infty \leqslant 1}$$

  where $\mathbf{Q} = \lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^n x_i x_i^\top \in \mathbb{R}^{p\times p}$ and $\mathbf{J} = \operatorname{Supp}(\mathbf{w})$

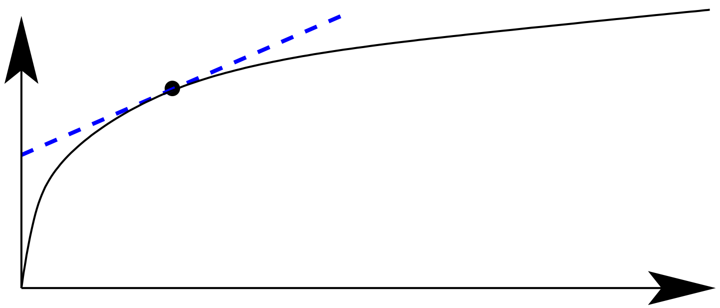
- **The Lasso is usually not model-consistent**
  - Selects more variables than necessary (see, e.g., Lv and Fan, 2009)
  - **Fixing the Lasso**: adaptive Lasso (Zou, 2006), relaxed Lasso (Meinshausen, 2008), thresholding (Lounici, 2008), Bolasso (Bach, 2008a), stability selection (Meinshausen and Bühlmann, 2008), Wasserman and Roeder (2009)

Adaptive Lasso and concave penalization

- **Adaptive Lasso** (Zou, 2006; Huang et al., 2008)
  - Weighted ℓ_1-norm: $\displaystyle \min_{w \in \mathbb{R}^p} L(w) + \lambda \sum_{j=1}^{p} \frac{|w_j|}{|\hat{w}_j|^\alpha}$
  - $\hat{w}$ estimator obtained from ℓ_2 or ℓ_1 regularization

- **Reformulation in terms of concave penalization**

$$\min_{w \in \mathbb{R}^p} L(w) + \sum_{j=1}^{p} g(|w_j|)$$



  - Example: $g(|w_j|) = |w_j|^{1/2}$ or $\log |w_j|$. Closer to the ℓ_0 penalty
  - Concave-convex procedure: replace $g(|w_j|)$ by affine upper bound
  - Better sparsity-inducing properties (Fan and Li, 2001; Zou and Li, 2008; Zhang, 2008b)

High-dimensional inference (Lasso)

- **Main result**: we only need $k \log p = O(n)$
  - if $\mathbf{w}$ is sufficiently sparse
  - **and** input variables are not too correlated

- Precise conditions on covariance matrix $\mathbf{Q} = \frac{1}{n}X^\top X$.
  - **Mutual incoherence** (Lounici, 2008)
  - **Restricted eigenvalue conditions** (Bickel et al., 2009)
  - Sparse eigenvalues (Meinshausen and Yu, 2008)
  - Null space property (Donoho and Tanner, 2005)

- Links with signal processing and compressed sensing (Candès and Wakin, 2008)

- Assume that $\mathbf{Q}$ has unit diagonal

Mutual incoherence (uniform low correlations)

- **Theorem** (Lounici, 2008):
  - $y_i = \mathbf{w}^\top x_i + \varepsilon_i$, ε i.i.d. normal with mean zero and variance σ^2
  - $\mathbf{Q} = X^\top X/n$ with unit diagonal and cross-terms less than $\dfrac{1}{14k}$
  - if $\|\mathbf{w}\|_0 \leqslant k$, and $A^2 > 8$, then, with $\lambda = A\sigma\sqrt{n \log p}$

$$\mathbb{P}\left(\|\hat{w} - \mathbf{w}\|_\infty \leqslant 5A\sigma \left(\frac{\log p}{n} \right)^{1/2} \right) \geqslant 1 - p^{1-A^2/8}$$

- Model consistency by thresholding if $\displaystyle\min_{j, \mathbf{w}_j \neq 0} |\mathbf{w}_j| > C\sigma\sqrt{\frac{\log p}{n}}$

- Mutual incoherence condition depends *strongly* on k

- Improved result by averaging over sparsity patterns (Candès and Plan, 2009)

Alternative sparse methods
Greedy methods

- Forward selection
- Forward-backward selection
- Non-convex method
  - Harder to analyze
  - Simpler to implement
  - Problems of stability
- Positive theoretical results (Zhang, 2009, 2008a)
  - Similar sufficient conditions than for the Lasso

Comparing Lasso and other strategies for linear regression

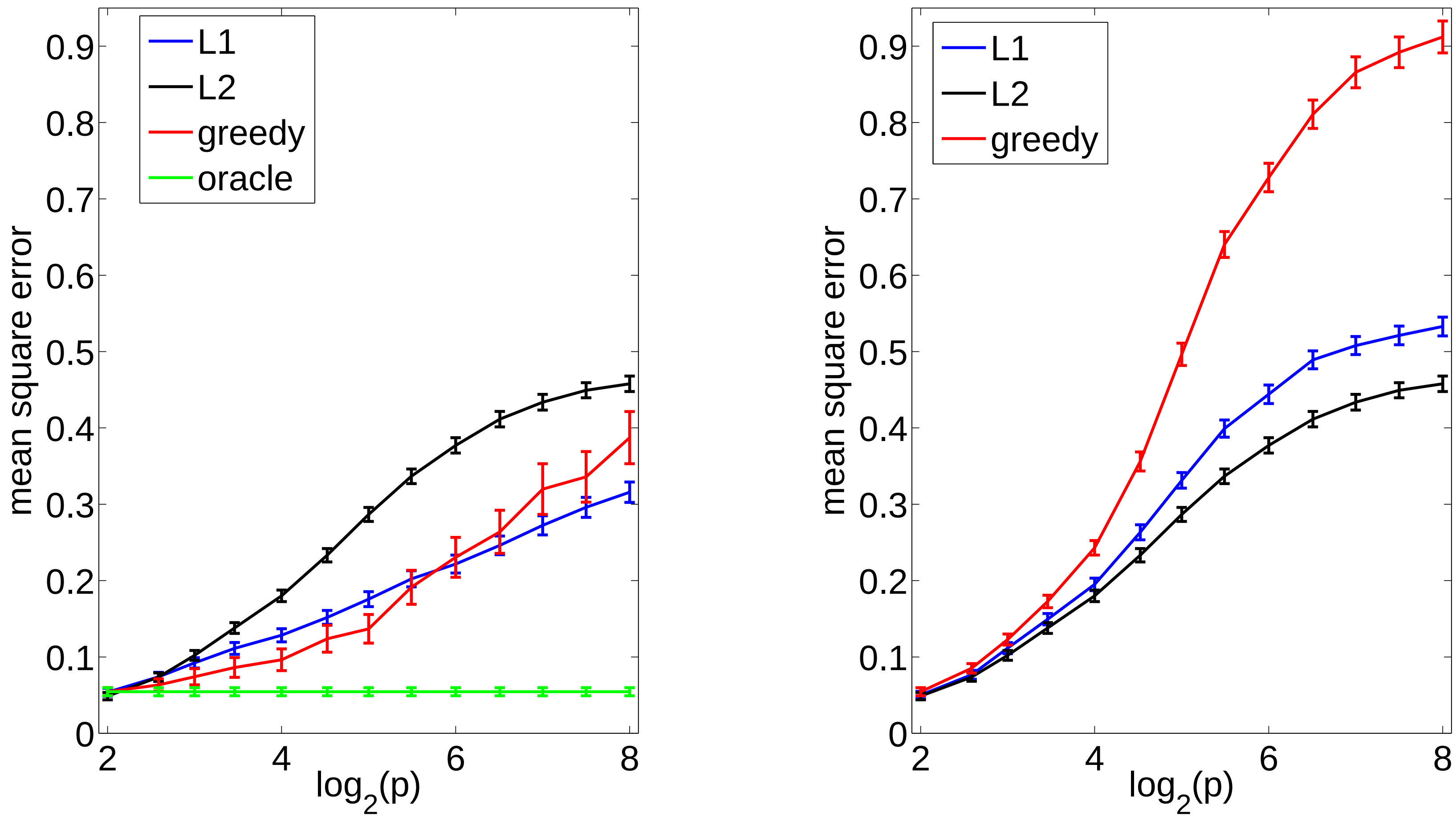
- Compared methods to reach the least-square solution
  - Ridge regression: $\displaystyle \min_{w\in\mathbb{R}^p} \frac{1}{2}\|y - Xw\|_2^2 + \frac{\lambda}{2}\|w\|_2^2$
  - Lasso: $\displaystyle \min_{w\in\mathbb{R}^p} \frac{1}{2}\|y - Xw\|_2^2 + \lambda\|w\|_1$
  - Forward greedy:
    - Initialization with empty set
    - Sequentially add the variable that best reduces the square loss

- Each method builds a path of solutions from 0 to ordinary least-squares solution

- Regularization parameters selected on the test set

Simulation results

- i.i.d. Gaussian design matrix, $k = 4$, $n = 64$, $p \in [2, 256]$, SNR $= 1$
- Note stability to non-sparsity and variability



Sparse	Rotated (non sparse)

Summary
ℓ_1-norm regularization

- ℓ_1-norm regularization leads to **nonsmooth optimization problems**
  - analysis through directional derivatives or subgradients
  - optimization may or may not take advantage of sparsity
- ℓ_1-norm regularization allows **high-dimensional inference**
- Interesting problems for ℓ_1-regularization
  - Stable variable selection
  - Weaker sufficient conditions (for weaker results)
  - Estimation of regularization parameter (all bounds depend on the unknown noise variance σ^2)

Extensions

- **Sparse methods are not limited to the square loss**
  - logistic loss: algorithms (Beck and Teboulle, 2009) and theory (Van De Geer, 2008; Bach, 2009)

- **Sparse methods are not limited to supervised learning**
  - Learning the structure of Gaussian graphical models (Meinshausen and Bühlmann, 2006; Banerjee et al., 2008)
  - Sparsity on matrices (last part of the tutorial)

- **Sparse methods are not limited to variable selection in a linear model**
  - **See next part of the tutorial**

Course outline

1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Penalization with grouped variables (Yuan and Lin, 2006)

- Assume that $\{1, \dots, p\}$ is **partitioned** into m groups $G_1, \dots, G_m$
- Penalization by $\sum_{i=1}^{m} \|w_{G_i}\|_2$, often called ℓ_1-ℓ_2 norm
- Induces group sparsity
  - Some groups entirely set to zero
  - no zeros within groups
- In this tutorial:
  - Groups may have infinite size $\Rightarrow$ **MKL**
  - Groups may overlap $\Rightarrow$ **structured sparsity** (Jenatton et al., 2009)

Linear vs. non-linear methods

- All methods in this tutorial are **linear in the parameters**
- By replacing x by features $\Phi(x)$, they can be made **non linear in the data**
- **Implicit vs. explicit features**
  - ℓ_1-norm: explicit features
  - ℓ_2-norm: representer theorem allows to consider implicit features if their dot products can be computed easily (kernel methods)

Kernel methods: regularization by ℓ_2-norm

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$, with **features** $\Phi(x) \in \mathcal{F} = \mathbb{R}^p$
  - Predictor $f(x) = w^\top \Phi(x)$ linear in the features

- Optimization problem: $\boxed{\displaystyle \min_{w \in \mathbb{R}^p} \sum_{i=1}^n \ell(y_i, w^\top \Phi(x_i)) + \frac{\lambda}{2} \|w\|_{\mathbf{2}}^2}$

Kernel methods: regularization by ℓ_2-norm

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$, with **features** $\Phi(x) \in \mathcal{F} = \mathbb{R}^p$
  - Predictor $f(x) = w^\top \Phi(x)$ linear in the features
- Optimization problem: $\boxed{\displaystyle \min_{w \in \mathbb{R}^p} \sum_{i=1}^n \ell(y_i, w^\top \Phi(x_i)) + \frac{\lambda}{2} \|w\|_{\mathbf{2}}^2}$
- **Representer theorem** (Kimeldorf and Wahba, 1971): solution must be of the form $w = \sum_{i=1}^n \alpha_i \Phi(x_i)$
  - Equivalent to solving: $\boxed{\displaystyle \min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i) + \frac{\lambda}{2} \alpha^\top K \alpha}$
  - Kernel matrix $K_{ij} = k(x_i, x_j) = \Phi(x_i)^\top \Phi(x_j)$

Multiple kernel learning (MKL) (Lanckriet et al., 2004b; Bach et al., 2004a)

- Sparse methods are linear!
- Sparsity with non-linearities
  - replace $f(x) = \sum_{j=1}^{p} w_j^\top x_j$ with $x \in \mathbb{R}^p$ and $w_j \in \mathbb{R}$
  - by $f(x) = \sum_{j=1}^{p} w_j^\top \Phi_j(x)$ with $x \in \mathcal{X}$, $\Phi_j(x) \in \mathcal{F}_j$ an $w_j \in \mathcal{F}_j$
- Replace the ℓ_1-norm $\sum_{j=1}^{p} |w_j|$ by "block" ℓ_1-norm $\sum_{j=1}^{p} \|w_j\|_2$
- Remarks
  - Hilbert space extension of the group Lasso (Yuan and Lin, 2006)
  - Alternative sparsity-inducing norms (Ravikumar et al., 2008)

Multiple kernel learning

- Learning combinations of kernels: $K(\eta) = \sum_{j=1}^{m} \eta_j K_j, \quad \eta \geqslant 0$
  - Summing kernels $\Leftrightarrow$ concatenating feature spaces
  - Assume $k_1(x,y) = \langle \Phi_1(x), \Phi_1(y) \rangle$, $k_2(x,y) = \langle \Phi_2(x), \Phi_2(y) \rangle$

$$k_1(x,y) + k_2(x,y) = \left\langle \binom{\Phi_1(x)}{\Phi_2(x)}, \binom{\Phi_1(y)}{\Phi_2(y)} \right\rangle$$

- Summing kernels $\Leftrightarrow$ generalized additive models

- Relationships with sparse additive models (Ravikumar et al., 2008)

Multiple kernel learning (MKL) (Lanckriet et al., 2004b; Bach et al., 2004a)

- Multiple feature maps / kernels on $x \in \mathcal{X}$:
  - p "feature maps" $\Phi_j : \mathcal{X} \mapsto \mathcal{F}_j$, $j = 1, \dots, p$.
  - Minimization with respect to $w_1 \in \mathcal{F}_1, \dots, w_p \in \mathcal{F}_p$
  - Predictor: $f(x) = w_1{}^\top \Phi_1(x) + \cdots + w_p{}^\top \Phi_p(x)$

$$
\begin{array}{ccccccc}
 & & \Phi_1(x)^\top & w_1 & & & \\
 & \nearrow & \vdots & \vdots & \searrow & & \\
x & \longrightarrow & \Phi_j(x)^\top & w_j & \longrightarrow & w_1^\top \Phi_1(x) + \cdots + w_p^\top \Phi_p(x) \\
 & \searrow & \vdots & \vdots & \nearrow & & \\
 & & \Phi_p(x)^\top & w_p & & &
\end{array}
$$

  - Generalized additive models (Hastie and Tibshirani, 1990)

Regularization for multiple features

$$
\begin{array}{ccccc}
 & & \Phi_1(x)^\top \quad w_1 & & \\
 & \nearrow & \vdots \qquad \vdots & \searrow & \\
x & \longrightarrow & \Phi_j(x)^\top \quad w_j & \longrightarrow & w_1^\top \Phi_1(x) + \cdots + w_p^\top \Phi_p(x) \\
 & \searrow & \vdots \qquad \vdots & \nearrow & \\
 & & \Phi_p(x)^\top \quad w_p & &
\end{array}
$$

- Regularization by $\sum_{j=1}^p \|w_j\|_2^2$ is equivalent to using $K = \sum_{j=1}^p K_j$
  - Summing kernels is equivalent to concatenating feature spaces

Regularization for multiple features

$$
\begin{array}{cccccc}
 & & \Phi_1(x)^\top & w_1 & & \\
 & \nearrow & \vdots & \vdots & \searrow & \\
x & \longrightarrow & \Phi_j(x)^\top & w_j & \longrightarrow & w_1^\top \Phi_1(x) + \cdots + w_p^\top \Phi_p(x) \\
 & \searrow & \vdots & \vdots & \nearrow & \\
 & & \Phi_p(x)^\top & w_p & &
\end{array}
$$

- Regularization by $\sum_{j=1}^{p} \|w_j\|_2^2$ is equivalent to using $K = \sum_{j=1}^{p} K_j$
- Regularization by $\sum_{j=1}^{p} \|w_j\|_2$ imposes sparsity at the group level
- **Main questions when regularizing by block ℓ_1-norm**:
  1. Algorithms
  2. Analysis of sparsity inducing properties (Ravikumar et al., 2008; Bach, 2008c)
  3. Does it correspond to a specific combination of kernels?

General kernel learning

- **Proposition** (Lanckriet et al, 2004, Bach et al., 2005, Micchelli and Pontil, 2005):

$$
\begin{aligned}
G(K) &= \min_{w \in \mathcal{F}} \textstyle\sum_{i=1}^n \ell(y_i, w^\top \Phi(x_i)) + \frac{\lambda}{2}\|w\|_2^2 \\
&= \max_{\alpha \in \mathbb{R}^n} -\textstyle\sum_{i=1}^n \ell_i^*(\lambda \alpha_i) - \frac{\lambda}{2}\alpha^\top K \alpha
\end{aligned}
$$

  is a **convex** function of the kernel matrix K

- Theoretical learning bounds (Lanckriet et al., 2004, Srebro and Ben-David, 2006)
  - Less assumptions than sparsity-based bounds, but slower rates

Equivalence with kernel learning (Bach et al., 2004a)

- Block ℓ_1-norm problem:

$$\sum_{i=1}^{n} \ell(y_i, w_1^\top \Phi_1(x_i) + \cdots + w_p^\top \Phi_p(x_i)) + \frac{\lambda}{2} (\|w_1\|_2 + \cdots + \|w_p\|_2)^2$$

- **Proposition**: Block ℓ_1-norm regularization is equivalent to minimizing with respect to η the optimal value $G(\sum_{j=1}^{p} \eta_j K_j)$

- (sparse) weights η obtained from optimality conditions

- dual parameters α optimal for $K = \sum_{j=1}^{p} \eta_j K_j$,

- **Single optimization problem for learning both η and α**

Proof of equivalence

$$
\begin{aligned}
& \min_{w_1,\dots,w_p} \sum_{i=1}^n \ell\Big(y_i, \sum_{j=1}^p w_j^\top \Phi_j(x_i)\Big) + \lambda\Big(\sum_{j=1}^p \|w_j\|_2\Big)^2 \\
= \quad & \min_{w_1,\dots,w_p} \min_{\sum_j \eta_j = 1} \sum_{i=1}^n \ell\Big(y_i, \sum_{j=1}^p w_j^\top \Phi_j(x_i)\Big) + \lambda \sum_{j=1}^p \|w_j\|_2^2/\eta_j \\
= \quad & \min_{\sum_j \eta_j = 1} \min_{\tilde{w}_1,\dots,\tilde{w}_p} \sum_{i=1}^n \ell\Big(y_i, \sum_{j=1}^p \eta_j^{1/2} \tilde{w}_j^\top \Phi_j(x_i)\Big) + \lambda \sum_{j=1}^p \|\tilde{w}_j\|_2^2 \text{ with } \tilde{w}_j = w_j \eta_j^{-1/2} \\
= \quad & \min_{\sum_j \eta_j = 1} \min_{\tilde{w}} \sum_{i=1}^n \ell\big(y_i, \tilde{w}^\top \Psi_\eta(x_i)\big) + \lambda \|\tilde{w}\|_2^2 \text{ with } \Psi_\eta(x) = (\eta_1^{1/2}\Phi_1(x), \dots, \eta_p^{1/2}\Phi_p(x))
\end{aligned}
$$

- We have: $\Psi_\eta(x)^\top \Psi_\eta(x') = \sum_{j=1}^p \eta_j k_j(x, x')$ with $\sum_{j=1}^p \eta_j = 1$ (and $\eta \geqslant 0$)

Algorithms for the group Lasso / MKL

- Group Lasso
  - Block coordinate descent (Yuan and Lin, 2006)
  - Active set method (Roth and Fischer, 2008; Obozinski et al., 2009)
  - Nesterov's accelerated method (Liu et al., 2009)

- MKL
  - Dual ascent, e.g., sequential minimal optimization (Bach et al., 2004a)
  - η-trick + cutting-planes (Sonnenburg et al., 2006)
  - η-trick + projected gradient descent (Rakotomamonjy et al., 2008)
  - Active set (Bach, 2008b)

Applications of multiple kernel learning

- **Selection of hyperparameters for kernel methods**
- **Fusion from heterogeneous data sources** (Lanckriet et al., 2004a)
- Two strategies for kernel combinations:
  - Uniform combination $\Leftrightarrow$ ℓ_2-norm
  - Sparse combination $\Leftrightarrow$ ℓ_1-norm
  - MKL always leads to more interpretable models
  - MKL does not always lead to better predictive performance
    * In particular, with few well-designed kernels
    * Be careful with normalization of kernels (Bach et al., 2004b)

Applications of multiple kernel learning

- **Selection of hyperparameters for kernel methods**
- **Fusion from heterogeneous data sources** (Lanckriet et al., 2004a)
- Two strategies for kernel combinations:
  - Uniform combination $\Leftrightarrow$ ℓ_2-norm
  - Sparse combination $\Leftrightarrow$ ℓ_1-norm
  - MKL always leads to more interpretable models
  - MKL does not always lead to better predictive performance
    - In particular, with few well-designed kernels
    - Be careful with normalization of kernels (Bach et al., 2004b)
- **Sparse methods**: new possibilities and new features

Course outline

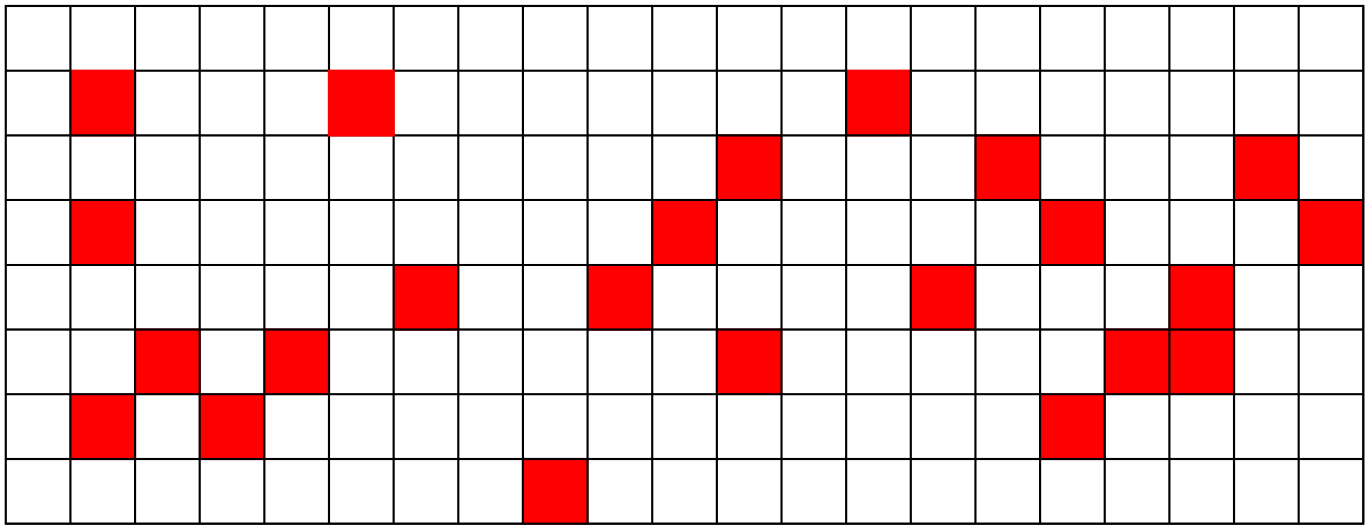
1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Learning on matrices - Image denoising

- Simultaneously denoise all patches of a given image
- Example from Mairal, Bach, Ponce, Sapiro, and Zisserman (2009)

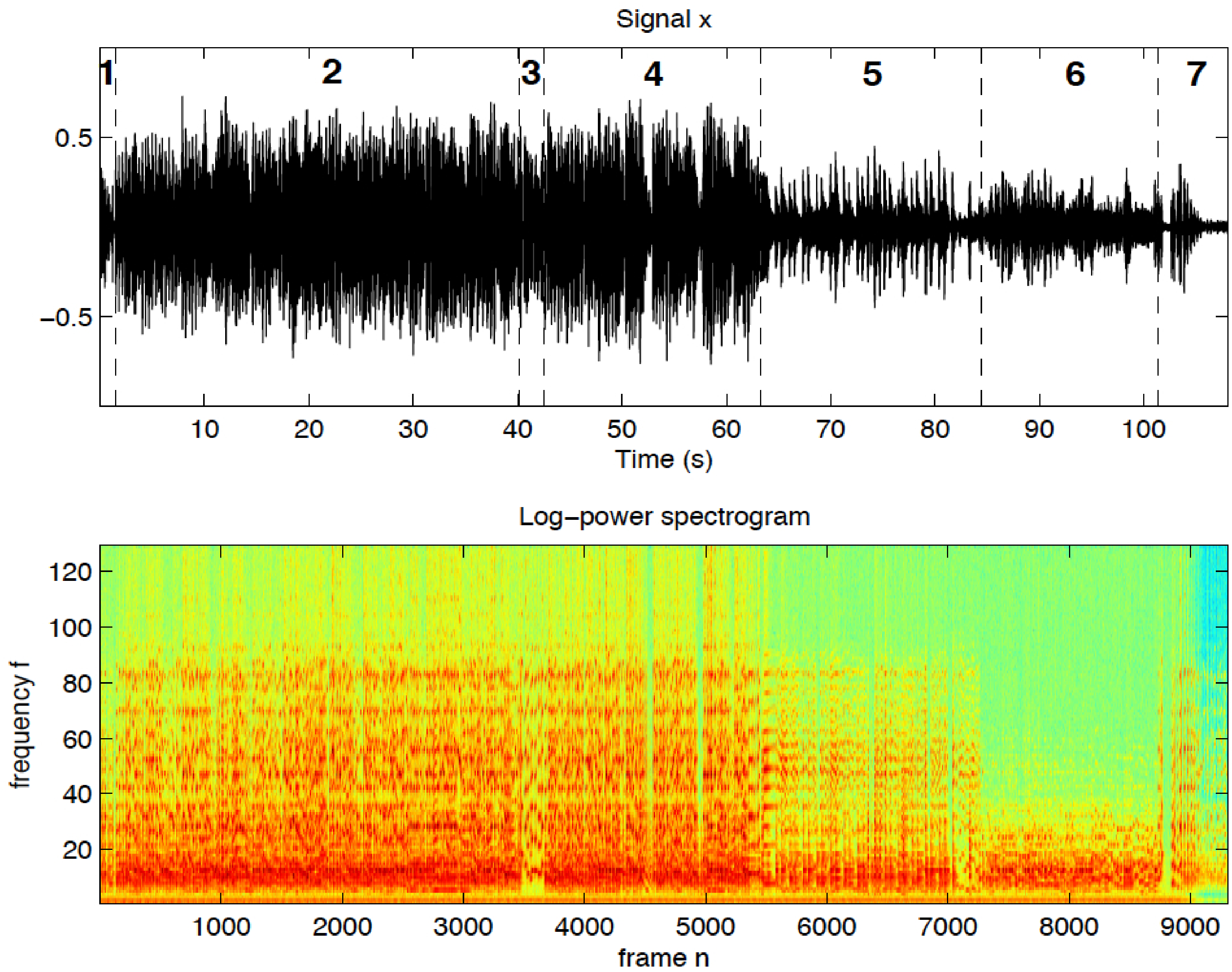

Learning on matrices - Collaborative filtering

- 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.



Learning on matrices - Source separation

- Single microphone (Benaroya et al., 2006; Févotte et al., 2009)

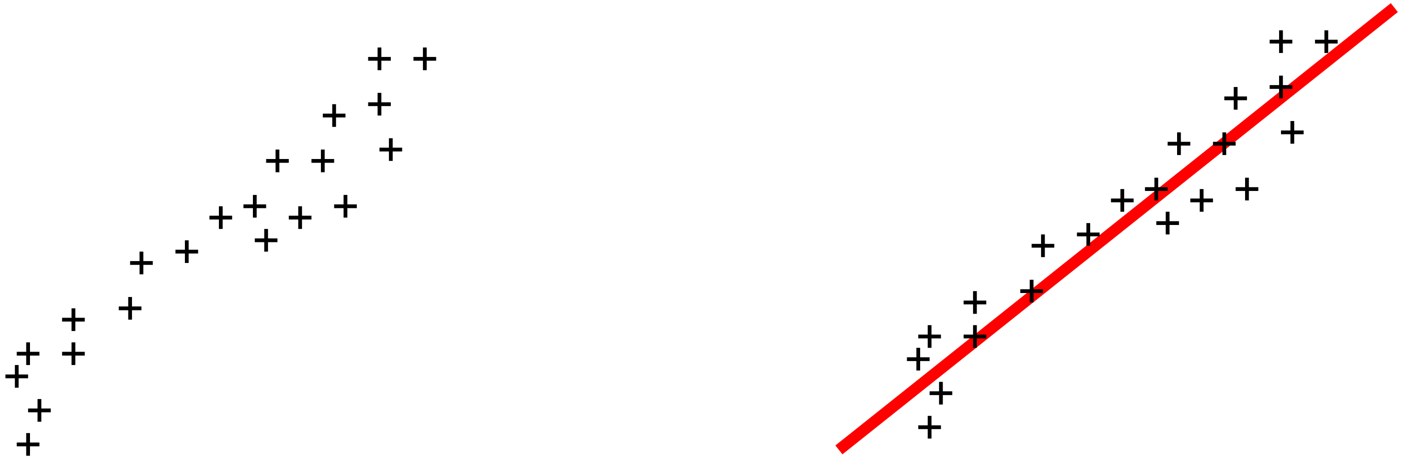


Learning on matrices - Multi-task learning

- k linear prediction tasks on same covariates $\mathbf{x} \in \mathbb{R}^p$
  - k weight vectors $\mathbf{w}_j \in \mathbb{R}^p$
  - Joint matrix of predictors $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathbb{R}^{p \times k}$
- Classical application
  - Multi-category classification (one task per class) (Amit et al., 2007)
- **Share parameters between tasks**
- **Joint variable selection** (Obozinski et al., 2009)
  - Select variables which are predictive for all tasks
- **Joint feature selection** (Pontil et al., 2007)
  - Construct linear features common to all tasks

Matrix factorization - Dimension reduction

- Given data matrix $\mathcal{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{p \times n}$
  - **Principal component analysis**: $\boxed{\mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i \Rightarrow \mathbf{X} = \mathbf{DA}}$

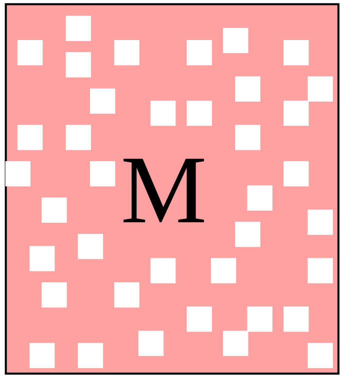


  - **K-means**: $\boxed{\mathbf{x}_i \approx \mathbf{d}_k \Rightarrow \mathbf{X} = \mathbf{DA}}$

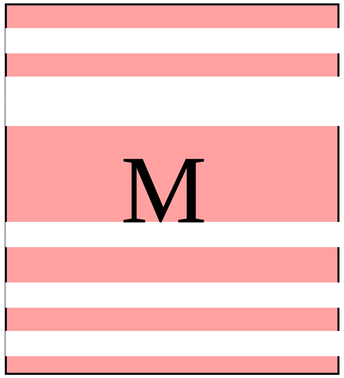
Two types of sparsity for matrices $\mathbf{M} \in \mathbb{R}^{n \times p}$

I - Directly on the elements of $\mathbf{M}$

- Many zero elements: $\mathbf{M}_{ij} = 0$



- Many zero rows (or columns): $(\mathbf{M}_{i1}, \ldots, \mathbf{M}_{ip}) = 0$

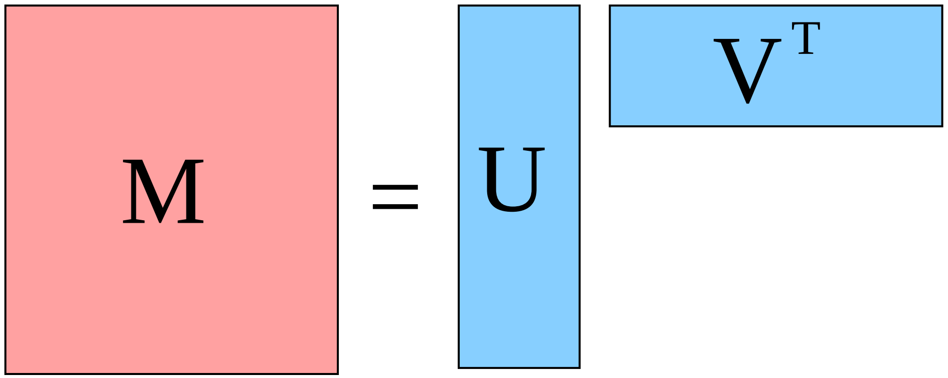


Two types of sparsity for matrices $\mathbf{M} \in \mathbb{R}^{n \times p}$

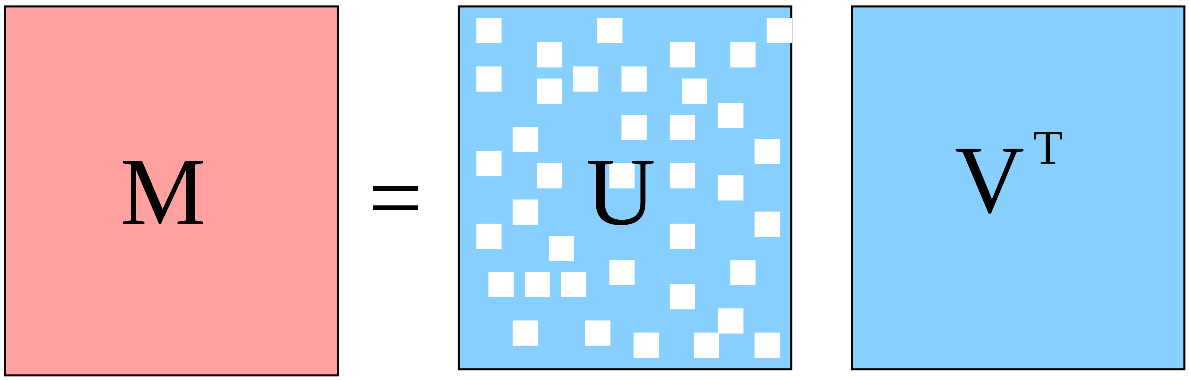
II - Through a factorization of $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$

- Matrix $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$, $\mathbf{U} \in \mathbb{R}^{n \times k}$ and $\mathbf{V} \in \mathbb{R}^{p \times k}$

- **Low rank**: m small



- **Sparse decomposition**: $\mathbf{U}$ sparse



Structured sparse matrix factorizations

- Matrix $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$, $\mathbf{U} \in \mathbb{R}^{n\times k}$ and $\mathbf{V} \in \mathbb{R}^{p\times k}$

- **Structure on $\mathbf{U}$ and/or $\mathbf{V}$**
  - Low-rank: $\mathbf{U}$ and $\mathbf{V}$ have few columns
  - Dictionary learning / sparse PCA: $\mathbf{U}$ has many zeros
  - Clustering (k-means): $\mathbf{U} \in \{0,1\}^{n\times m}$, $\mathbf{U}\mathbf{1} = \mathbf{1}$
  - Pointwise positivity: non negative matrix factorization (NMF)
  - Specific patterns of zeros (Jenatton et al., 2010)
  - Low-rank + sparse (Candès et al., 2009)
  - etc.

- **Many applications**

- **Many open questions** (Algorithms, identifiability, etc.)

Multi-task learning

- Joint matrix of predictors $W = (w_1, \ldots, w_k) \in \mathbb{R}^{p \times k}$
- **Joint variable selection** (Obozinski et al., 2009)
  - Penalize by the sum of the norms of rows of W (group Lasso)
  - Select variables which are predictive for all tasks

Multi-task learning

- Joint matrix of predictors $W = (w_1, \ldots, w_k) \in \mathbb{R}^{p \times k}$

- **Joint variable selection** (Obozinski et al., 2009)
  - Penalize by the sum of the norms of rows of W (group Lasso)
  - Select variables which are predictive for all tasks

- **Joint feature selection** (Pontil et al., 2007)
  - Penalize by the trace-norm (see later)
  - Construct linear features common to all tasks

- Theory: allows number of observations which is sublinear in the number of tasks (Obozinski et al., 2008; Lounici et al., 2009)

- Practice: more interpretable models, slightly improved performance

Low-rank matrix factorizations
Trace norm

- Given a matrix $\mathbf{M} \in \mathbb{R}^{n \times p}$
  - Rank of $\mathbf{M}$ is the minimum size m of **all** factorizations of $\mathbf{M}$ into $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$, $\mathbf{U} \in \mathbb{R}^{n \times m}$ and $\mathbf{V} \in \mathbb{R}^{p \times m}$
  - Singular value decomposition: $\mathbf{M} = \mathbf{U}\,\mathrm{Diag}(\mathbf{s})\mathbf{V}^\top$ where $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns and $\mathbf{s} \in \mathbb{R}_+^m$ are singular values

- Rank of $\mathbf{M}$ equal to the number of non-zero singular values

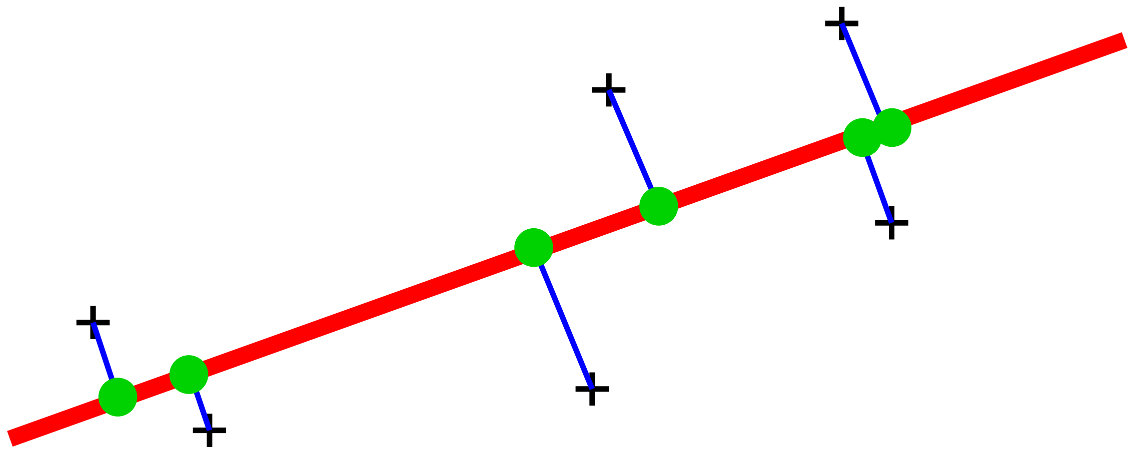
Low-rank matrix factorizations
Trace norm

- Given a matrix $\mathbf{M} \in \mathbb{R}^{n \times p}$
  - Rank of $\mathbf{M}$ is the minimum size m of **all** factorizations of $\mathbf{M}$ into $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$, $\mathbf{U} \in \mathbb{R}^{n \times m}$ and $\mathbf{V} \in \mathbb{R}^{p \times m}$
  - Singular value decomposition: $\mathbf{M} = \mathbf{U}\,\mathrm{Diag}(\mathbf{s})\mathbf{V}^\top$ where $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns and $\mathbf{s} \in \mathbb{R}^m_+$ are singular values
- Rank of $\mathbf{M}$ equal to the number of non-zero singular values
- **Trace-norm (a.k.a. nuclear norm)** $=$ sum of singular values
- Convex function, leads to a semi-definite program (Fazel et al., 2001)
- First used for collaborative filtering (Srebro et al., 2005)

Sparse principal component analysis

- Given data $\mathcal{X} = (\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top) \in \mathbb{R}^{p \times n}$, two views of PCA:
  - **Analysis view**: find the projection $\mathbf{d} \in \mathbb{R}^p$ of maximum variance (with deflation to obtain more components)
  - **Synthesis view**: find the basis $\mathbf{d}_1, \dots, \mathbf{d}_k$ such that all $\mathbf{x}_i$ have low reconstruction error when decomposed on this basis

- For regular PCA, the two views are equivalent



Sparse principal component analysis

- Given data $\mathcal{X} = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top) \in \mathbb{R}^{p \times n}$, two views of PCA:
  - **Analysis view**: find the projection $\mathbf{d} \in \mathbb{R}^p$ of maximum variance (with deflation to obtain more components)
  - **Synthesis view**: find the basis $\mathbf{d}_1, \ldots, \mathbf{d}_k$ such that all $\mathbf{x}_i$ have low reconstruction error when decomposed on this basis

- For regular PCA, the two views are equivalent

- **Sparse extensions**
  - Interpretability
  - High-dimensional inference
  - Two views are differents
    - For analysis view, see d'Aspremont, Bach, and El Ghaoui (2008)

Sparse principal component analysis
Synthesis view

- Find $\mathbf{d}_1, \ldots, \mathbf{d}_k \in \mathbb{R}^p$ **sparse** so that

$$\sum_{i=1}^{n} \min_{\boldsymbol{\alpha}_i \in \mathbb{R}^m} \left\| \mathbf{x}_i - \sum_{j=1}^{k} (\boldsymbol{\alpha}_i)_j \mathbf{d}_j \right\|_2^2 = \sum_{i=1}^{n} \min_{\boldsymbol{\alpha}_i \in \mathbb{R}^m} \left\| \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i \right\|_2^2 \text{ is small}$$

  – Look for $\mathbf{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n) \in \mathbb{R}^{k \times n}$ and $\mathbf{D} = (\mathbf{d}_1, \ldots, \mathbf{d}_k) \in \mathbb{R}^{p \times k}$ such that $\mathbf{D}$ is sparse and $\|\mathcal{X} - \mathbf{D}\mathbf{A}\|_F^2$ is small

Sparse principal component analysis
Synthesis view

- Find $\mathbf{d}_1, \dots, \mathbf{d}_k \in \mathbb{R}^p$ **sparse** so that

$$\sum_{i=1}^{n} \min_{\boldsymbol{\alpha}_i \in \mathbb{R}^m} \left\| \mathbf{x}_i - \sum_{j=1}^{k} (\boldsymbol{\alpha}_i)_j \mathbf{d}_j \right\|_2^2 = \sum_{i=1}^{n} \min_{\boldsymbol{\alpha}_i \in \mathbb{R}^m} \left\| \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i \right\|_2^2 \text{ is small}$$

  – Look for $\mathbf{A} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_n) \in \mathbb{R}^{k \times n}$ and $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_k) \in \mathbb{R}^{p \times k}$ such that $\mathbf{D}$ is sparse and $\|\mathcal{X} - \mathbf{DA}\|_F^2$ is small

- Sparse formulation (Witten et al., 2009; Bach et al., 2008)

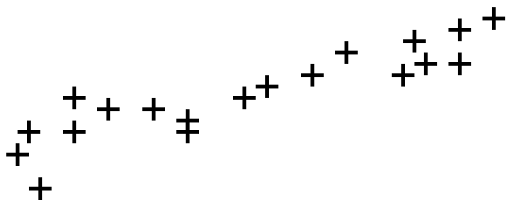
  – Penalize/constrain $\mathbf{d}_j$ by the ℓ_1-norm for sparsity

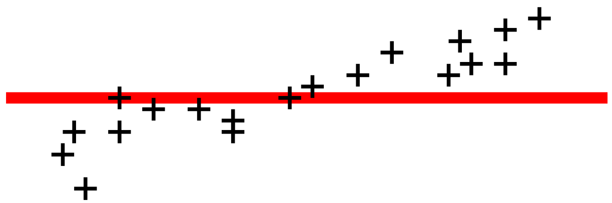
  – Penalize/constrain $\boldsymbol{\alpha}_i$ by the ℓ_2-norm to avoid trivial solutions

$$\min_{\mathbf{D},\mathbf{A}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{j=1}^{k} \|\mathbf{d}_j\|_1 \text{ s.t. } \forall i, \|\boldsymbol{\alpha}_i\|_2 \leqslant 1$$

Sparse PCA vs. dictionary learning

- **Sparse PCA**: $\mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i$, $\mathbf{D}$ sparse

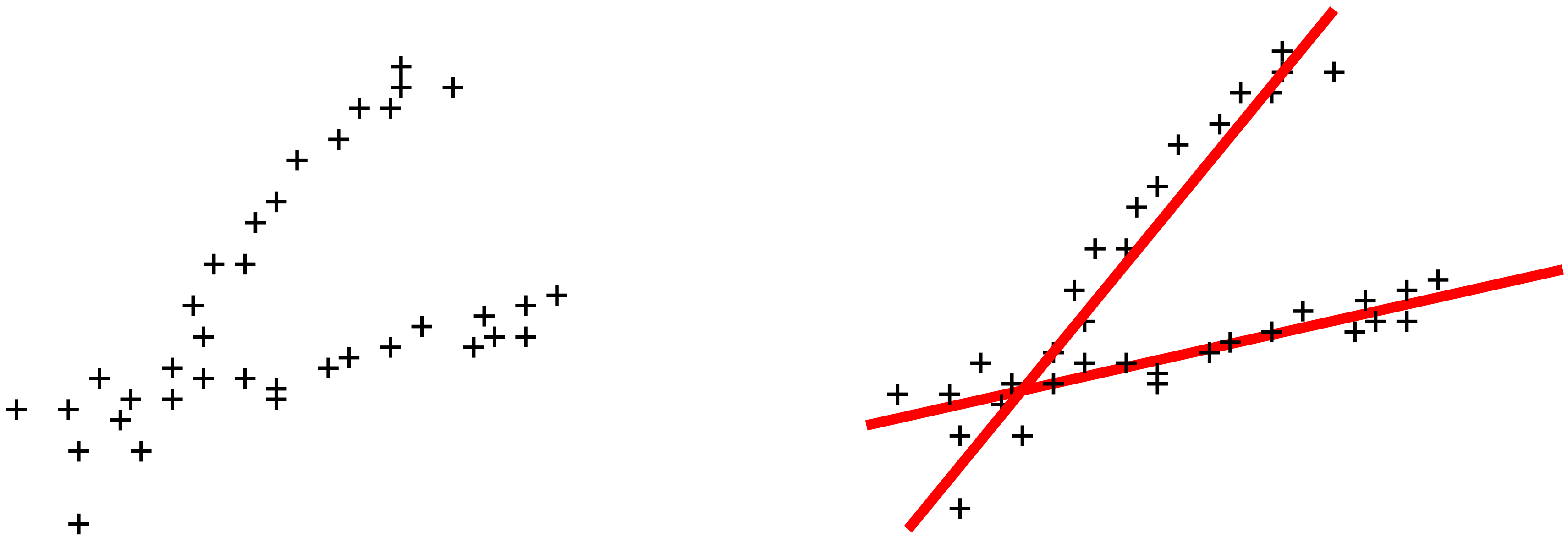




Sparse PCA vs. dictionary learning

- **Sparse PCA**: $\mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i$, $\mathbf{D}$ sparse

- **Dictionary learning**: $\mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i$, $\boldsymbol{\alpha}_i$ sparse



Structured matrix factorizations (Bach et al., 2008)

$$\min_{\mathbf{D},\mathbf{A}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{j=1}^{k} \|\mathbf{d}_j\|_{\star} \text{ s.t. } \forall i, \|\boldsymbol{\alpha}_i\|_{\bullet} \leqslant 1$$

$$\min_{\mathbf{D},\mathbf{A}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{i=1}^{n} \|\boldsymbol{\alpha}_i\|_{\bullet} \text{ s.t. } \forall j, \|\mathbf{d}_j\|_{\star} \leqslant 1$$

- Optimization by alternating minimization (non-convex)
- $\boldsymbol{\alpha}_i$ decomposition coefficients (or "code"), $\mathbf{d}_j$ dictionary elements
- Two related/equivalent problems:
  - **Sparse PCA = sparse dictionary** (ℓ_1-norm on $\mathbf{d}_j$)
  - **Dictionary learning = sparse decompositions** (ℓ_1-norm on $\boldsymbol{\alpha}_i$) (Olshausen and Field, 1997; Elad and Aharon, 2006; Lee et al., 2007)

Dictionary learning for image denoising

$$\underbrace{\mathbf{x}}_{\text{measurements}} = \underbrace{\mathbf{y}}_{\text{original image}} + \underbrace{\varepsilon}_{\text{noise}}$$

Sparse methods for machine learning
Why use sparse methods?

- **Sparsity as a proxy to interpretability**
  - Structured sparsity
- **Sparsity for high-dimensional inference**
  - Influence on feature design
- **Sparse methods are not limited to least-squares regression**
- **Faster training/testing**
- **Better predictive performance?**
  - Problems are sparse if you look at them the right way

Conclusion - Interesting questions/issues

- Implicit vs. explicit features
  - Can we algorithmically achieve $\log p = O(n)$ with explicit unstructured features?
- Norm design
  - What type of behavior may be obtained with sparsity-inducing norms?
- Overfitting convexity
  - Do we actually need convexity for matrix factorization problems?

Course outline

1. **Losses for particular machine learning tasks**
   - Classification, regression, etc...
2. **Regularization by Hilbert norms (kernel methods)**
   - Kernels and representer theorem
   - Convex duality, optimization and algorithms
   - Kernel methods
   - Kernel design
3. **Regularization by sparsity-inducing norms**
   - ℓ_1-norm regularization
   - Multiple kernel learning
   - Theoretical results
   - Learning on matrices

Conclusion - Interesting problems (machine learning)

- Kernel design for computer vision
  - Benefits of "kernelizing" existing representations
  - Combining kernels
- Sparsity and computer vision
  - Going beyond image denoising
- Large numbers of classes
  - Theoretical and algorithmic challenges
- Structured output

References

Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th international conference on Machine Learning (ICML)*, 2007.

N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, 68:337–404, 1950.

F. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML)*, 2008a.

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Adv. NIPS*, 2008b.

F. Bach. Self-concordant analysis for logistic regression. Technical Report 0910.4627, ArXiv, 2009.

F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. Technical Report 0812.1869, ArXiv, 2008.

F. R. Bach. Consistency of the group Lasso and multiple kernel learning. *Journal of Machine Learning Research*, pages 1179–1225, 2008c.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2004a.

F. R. Bach, R. Thibaux, and M. I. Jordan. Computing regularization paths for learning multiple kernels. In *Advances in Neural Information Processing Systems 17*, 2004b.

F. R. Bach, D. Heckerman, and E. Horvitz. Considering cost asymmetry in learning classifiers. *Journal of Machine Learning Research*, 7:1713–1741, 2006.

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, 9: 485–516, 2008.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Speech and Audio Processing*, 14(1):191, 2006.

D. Bertsekas. *Nonlinear programming*. Athena Scientific, 1995.

P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, 37(4):1705–1732, 2009.

Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21, 2005.

L. Bottou and C. J. Lin. Support vector machine solvers. In *Large scale kernel machines*, 2007.

Léon Bottou and Olivier Bousquet. Learning using large datasets. In *Mining Massive DataSets for Security*, NATO ASI Workshop Series. IOS Press, Amsterdam, 2008. URL `http://leon.bottou.org/papers/bottou-bousquet-2008b`. to appear.

E.J. Candès and Y. Plan. Near-ideal model selection by l1 minimization. *The Annals of Statistics*, 37 (5A):2145–2177, 2009.

E.J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Arxiv preprint arXiv:0912.3599*, 2009.

Emmanuel Candès and Michael Wakin. An introduction to compressive sampling. *IEEE Signal*

Processing Magazine, 25(2):21–30, 2008.

O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001. ISSN 0036-1445.

A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

D.L. Donoho and J. Tanner. Neighborliness of randomly projected simplices in high dimensions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9452, 2005.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32:407, 2004.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Proc.*, 15(12):3736–3745, 2006.

J. Fan and R. Li. Variable Selection Via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of the American Statistical Association*, 96(456):1348–1361, 2001.

M. Fazel, H. Hindi, and S.P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739, 2001.

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence. with application to music analysis. *Neural Computation*, 21(3), 2009.

S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of*

Machine Learning Research, 2:243–264, 2001.

P. A. Flach. The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In *International Conference on Machine Learning (ICML)*, 2003.

W. Fu. Penalized regressions: the bridge vs. the Lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998).

Thomas Gärtner, Peter A. Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *COLT*, 2003.

K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *J. Mach. Learn. Res.*, 8:725–760, 2007. ISSN 1533-7928.

Z. Harchaoui and F. R. Bach. Image classification with segmentation graph kernels. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2005.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Chapman & Hall, 1990.

David Haussler. Convolution kernels on discrete structures. Technical report, UCSC, 1999.

M. Hein and O. Bousquet. Hilbertian metrics and positive-definite kernels on probability measures. In *AISTATS*, 2004.

R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 2000.

J. Huang, S. Ma, and C.H. Zhang. Adaptive Lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18:1603–1618, 2008.

R. Jenatton, J.Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, arXiv:0904.3523, 2009.

R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *Submitted to ICML*, 2010.

T. Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

T. Joachims. Making large-scale support vector machine learning practical. In *Advances in kernel methods — Support Vector learning*. MIT Press, 1998.

Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Kernels for graphs. In *Kernel Methods in Computational Biology*. MIT Press, 2004.

G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applicat.*, 33:82–95, 1971.

G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinf.*, 20:2626–2635, 2004a.

G. R. G. Lanckriet, N. Cristianini, L. El Ghaoui, P. Bartlett, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004b.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *NIPS*, 2007.

J. Liu, S. Ji, and J. Ye. Multi-Task Feature Learning Via Efficient l2,-Norm Minimization. *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

G. Loosli, S. Canu, S. Vishwanathan, A. Smola, and M. Chattopadhyay. Boîte à outils SVM simple et rapide. *Revue dIntelligence Artificielle*, 19(4-5):741–767, 2005.

K. Lounici. Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators. *Electronic Journal of Statistics*, 2, 2008.

K. Lounici, A.B. Tsybakov, M. Pontil, and S.A. van de Geer. Taking advantage of sparsity in multi-task learning. In *Conference on Computational Learning Theory (COLT)*, 2009.

J. Lv and Y. Fan. A unified approach to model selection and sparse recovery using regularized least squares. *Annals of Statistics*, 37(6A):3498–3528, 2009.

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *International Conference on Computer Vision (ICCV)*, 2009.

H. M. Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, 3:111–133, 1956.

N. Meinshausen. Relaxed Lasso. *Computational Statistics and Data Analysis*, 52(1):374–393, 2008.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of statistics*, 34(3):1436, 2006.

N. Meinshausen and P. Bühlmann. Stability selection. Technical report, arXiv: 0809.2932, 2008.

N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, 37(1):246–270, 2008.

F. Meyer. Hierarchies of partitions and morphological segmentation. In *Scale-Space and Morphology in Computer Vision*. Springer-Verlag, 2001.

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Pub, 2003.

Y. Nesterov. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Tech. Rep*, 76, 2007.

G. Obozinski, M.J. Wainwright, and M.I. Jordan. High-dimensional union support recovery in multivariate regression. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

G. Obozinski, B. Taskar, and M.I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, pages 1–22, 2009.

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: Support Vector Learning*, 1998.

M. Pontil, A. Argyriou, and T. Evgeniou. Multi-task feature learning. In *Advances in Neural Information Processing Systems*, 2007.

A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. Simplemkl. *Journal of Machine Learning Research*, to appear, 2008.

Jan Ramon and Thomas Gärtner. Expressivity versus efficiency of graph kernels. In *First International Workshop on Mining Graphs, Trees and Sequences*, 2003.

P. Ravikumar, H. Liu, J. Lafferty, and L. Wasserman. SpAM: Sparse additive models. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Ann. Statist.*, 35(3):1012–1030, 2007.

V. Roth and B. Fischer. The group-Lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2001.

B. Schölkopf, J. C. Platt, J. S. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2007.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Camb. U. P., 2004.

S. Sonnenburg, G. Räsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, 2005.

B. Taskar. Structured prediction: A large margin approach. In *NIPS Tutorial*, 2005. URL `media.nips.cc/Conferences/2007/Tutorials/Slides/taskar-NIPS-07-tutorial.ppt`.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of The Royal Statistical Society Series B*, 58(1):267–288, 1996.

S. A. Van De Geer. High-dimensional generalized linear models and the Lasso. *Annals of Statistics*, 36 (2):614, 2008.

R. Vert and J.-P. Vert. Consistency and convergence rates of one-class svms and related algorithms. *Journal of Machine Learning Research*, 7:817–854, 2006.

S. V. N. Vishwanathan, A. J. Smola, and M. Murty. Simplesvm. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

M. J. Wainwright. Sharp thresholds for noisy and high-dimensional recovery of sparsity using ℓ_1-constrained quadratic programming. Technical Report 709, Dpt. of Statistics, UC Berkeley, 2006.

L. Wasserman and K. Roeder. High dimensional variable selection. *Annals of statistics*, 37(5A):2178, 2009.

C. Watkins. Dynamic alignment kernels. Technical report, RHUL, 1999.

D.M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.*, 2(1):224–244, 2008.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of The Royal Statistical Society Series B*, 68(1):49–67, 2006.

M. Yuan and Y. Lin. On the non-negative garrotte estimator. *Journal of The Royal Statistical Society Series B*, 69(2):143–161, 2007.

T. Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. *Advances in Neural Information Processing Systems*, 22, 2008a.

T. Zhang. Multi-stage convex relaxation for learning with sparse regularization. *Advances in Neural Information Processing Systems*, 22, 2008b.

T. Zhang. On the consistency of feature selection using greedy least squares regression. *The Journal* *of Machine Learning Research*, 10:555–568, 2009.

P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

H. Zou. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, 36(4):1509–1533, 2008.

Code

- SVM and other supervised learning techniques
  `www.shogun-toolbox.org`
  `http://gaelle.loosli.fr/research/tools/simplesvm.html`
  `http://www.kyb.tuebingen.mpg.de/bs/people/spider/main.html`

- ℓ^1-penalization: Matlab/C/R codes available from

  – SPAMS (SPArse Modeling Software)
    `http://www.di.ens.fr/willow/SPAMS/`

- Multiple kernel learning:
  `asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html`
  `www.stat.berkeley.edu/~gobo/SKMsmo.tar`