Learning with sparsity-inducing norms

Francis Bach

INRIA - Ecole Normale Supérieure

MLSS 2008 - Ile de Ré, 2008

Supervised learning and regularization

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

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

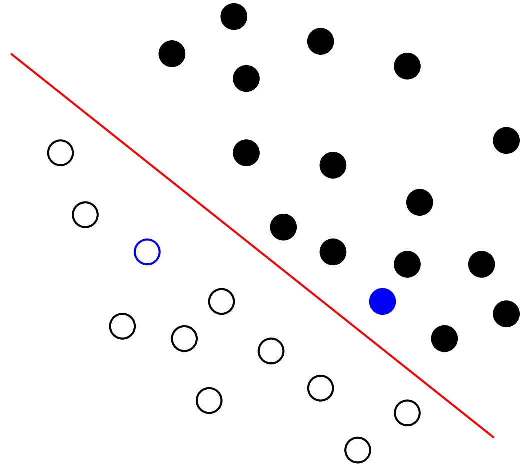
Error on data + Regularization

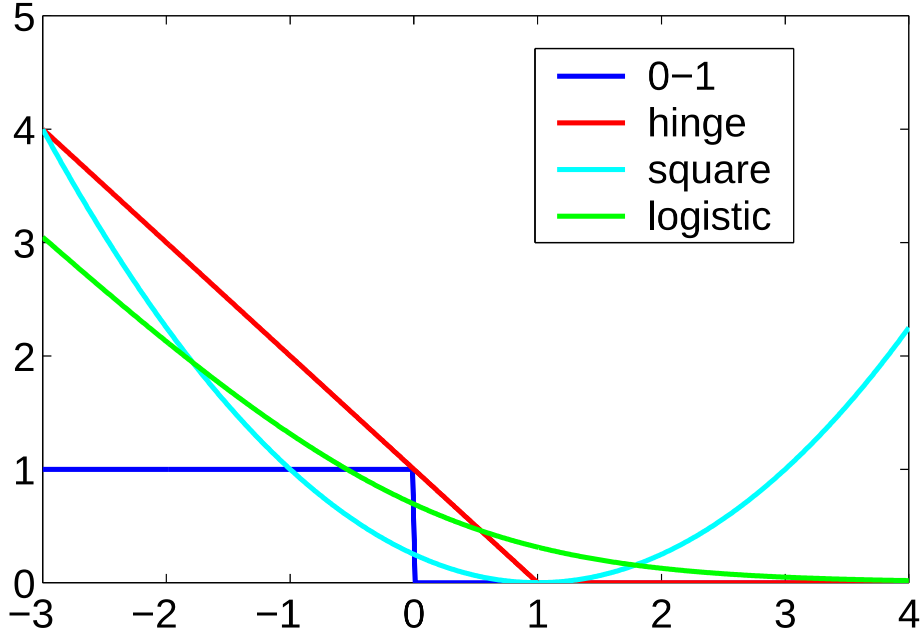
Loss & function space ? Norm ?

- Two issues:
  - Loss
  - Function space / norm

Usual losses [SS01, STC04]

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





Regularizations

- Main goal: control the "capacity" of the learning problem

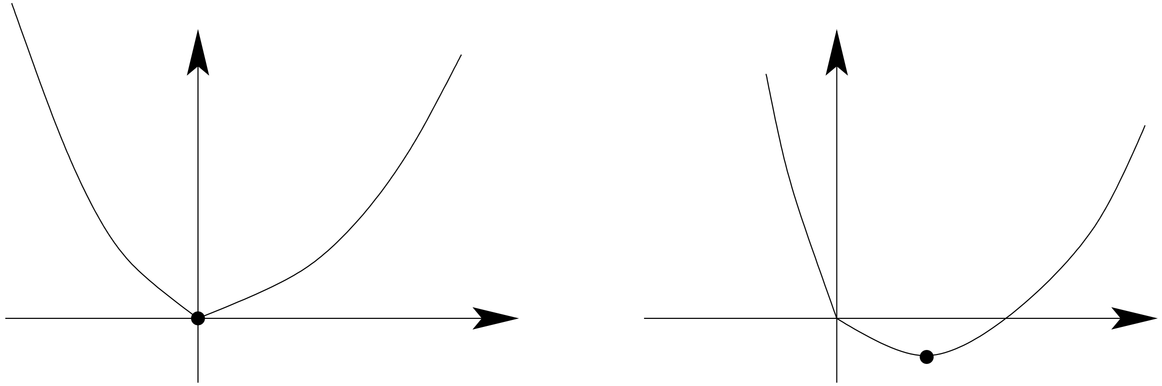
- Two main lines of work

  1. Use Hilbertian (RKHS) norms
     - Non parametric supervised learning and kernel methods
     - Well developped theory [SS01, STC04, Wah90]
  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
     - Often used heuristically

- **Goal of the course: Understand how and when to use sparsity-inducing norms**

Why ℓ_1-norms lead to sparsity?

- Example 1: quadratic problem in 1D, i.e. $\boxed{\min_{x\in\mathbb{R}} \frac{1}{2}x^2 - xy + \lambda|x|}$
- Piecewise quadratic function with a kink at zero
  - Derivative at $0+$: $g_+ = \lambda - y$ and $0-$: $g_- = -\lambda - y$



  - $x = 0$ is the solution iff $g_+ \geqslant 0$ and $g_- \leqslant 0$ (i.e., $|y| \leqslant \lambda$)
  - $x \geqslant 0$ is the solution iff $g_+ \leqslant 0$ (i.e., $y \geqslant \lambda$) $\Rightarrow x^* = y - \lambda$
  - $x \leqslant 0$ is the solution iff $g_- \leqslant 0$ (i.e., $y \leqslant -\lambda$) $\Rightarrow x^* = y + \lambda$

- Solution $\boxed{x^* = \mathrm{sign}(y)(|y| - \lambda)_+}$ = soft thresholding

Why ℓ_1-norms lead to sparsity?

- Example 2: isotropic quadratic problem

- $\displaystyle \min_{x \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{p} x_i^2 - \sum_{i=1}^{p} x_i y_i + \lambda \|x\|_1 = \min_{x \in \mathbb{R}^p} \frac{1}{2} x^\top x - x^\top y + \lambda \|x\|_1$

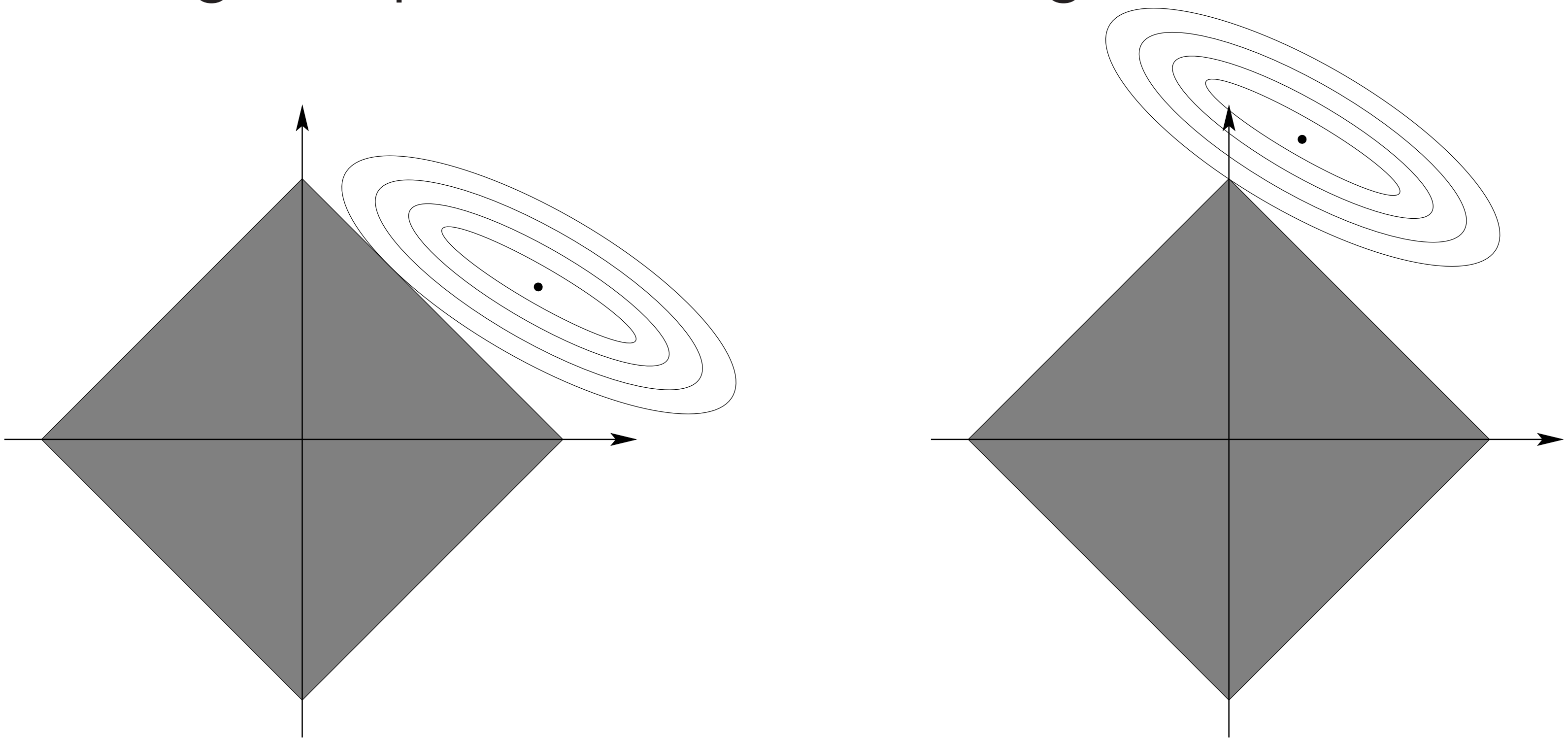
- solution: $x_i^* = \mathrm{sign}(y_i)(|y_i| - \lambda)_+$

- decoupled soft thresholding

Why ℓ_1-norms lead to sparsity?

- Example 3: general quadratic problem
  - coupled soft thresolding

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



Course Outline

1. ℓ^1**-norm regularization**
   - Review of nonsmooth optimization problems and algorithms
   - Algorithms for the Lasso (generic or dedicated)
   - Examples

2. **Extensions**
   - Group Lasso and multiple kernel learning (MKL) $+$ case study
   - Sparse methods for matrices
   - Sparse PCA

3. **Theory - Consistency of pattern selection**
   - Low and high dimensional setting
   - Links with compressed sensing

ℓ_1-norm regularization

- Data: covariates $x_i \in \mathbb{R}^p$, responses $y_i \in \mathcal{Y}$, $i = 1, \dots, n$, given in vector $y \in \mathbb{R}^p$ and matrix $X \in \mathbb{R}^{n \times p}$

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

$$\sum_{i=1}^{n} \ell(y_i, w^\top x_i) \quad + \quad \lambda \|w\|_1$$

$$\text{Error on data} \quad + \quad \text{Regularization}$$

- Including a constant term b?

- Assumptions on loss:
  - **convex** and **differentiable** in the second variable
  - NB: with the square loss $\Rightarrow$ basis pursuit (signal processing) [CDS01], Lasso (statistics/machine learning) [Tib96]

A review of nonsmooth convex analysis and optimization

- Analysis: optimality conditions
- Optimization: algorithms
  - First order methods
  - Second order methods
- Books: Boyd & VandenBerghe [BV03], Bonnans et al.[BGLS03], Nocedal & Wright [NW06], Borwein & Lewis [BL00]

Optimality conditions for ℓ^1-norm regularization

- Convex differentiable problems $\Rightarrow$ zero gradient!
  - Example: ℓ^2-regularization, i.e., $\min_w \sum_{i=1}^n \ell(y_i, w^\top x_i) + \frac{\lambda}{2} w^\top w$
  - Gradient $= \sum_{i=1}^n \ell'(y_i, w^\top x_i) x_i + \lambda w$ where $\ell'(y_i, w^\top x_i)$ is the partial derivative of the loss w.r.t the second variable
  - If square loss, $\sum_{i=1}^n \ell(y_i, w^\top x_i) = \frac{1}{2}\|y - Xw\|_2^2$ and gradient $= -X^\top (y - Xw) + \lambda w$

    $\Rightarrow$ normal equations $\Rightarrow w = (X^\top X + \lambda I)^{-1} X^\top Y$

- ℓ^1-norm is non differentiable!
  - How to compute the gradient of the absolute value?

- WARNING - gradient methods on non smooth problems! - WARNING

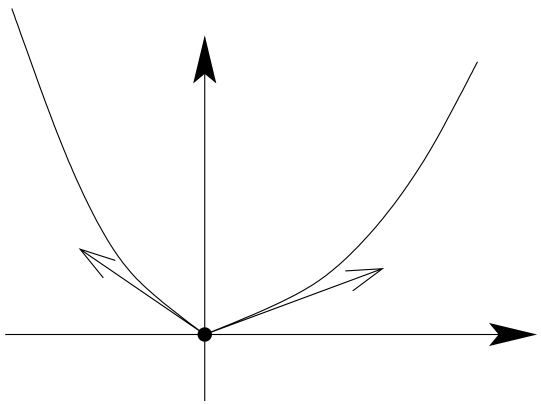
$\Rightarrow$ Directional derivatives - subgradient

Directional derivatives

- Directional derivative in the direction Δ at w:

$$\nabla J(w, \Delta) = \lim_{\varepsilon \to 0+} \frac{J(w + \varepsilon\Delta) - J(w)}{\varepsilon}$$

- Main idea: in non smooth situations, may need to look at all directions Δ and not simply p independent ones!



- **Proposition**: J is differentiable at w, if $\Delta \mapsto \nabla J(w, \Delta)$ is then linear, and $\nabla J(w, \Delta) = \nabla J(w)^\top \Delta$

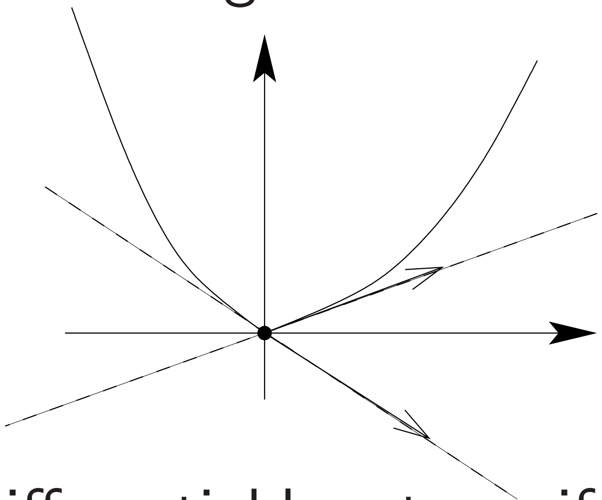
Subgradient

- Generalization of gradients for non smooth functions

- Definition: g is a subgradient of J at w if and only if

$$\forall t \in \mathbb{R}^p, \;\; J(t) \geqslant J(w) + g^\top (t - w)$$

(i.e., slope of lower bounding affine function)



- **Proposition**: J differentiable at w if and only if exactly one subgradient (the gradient)

- **Proposition**: (proper) convex functions always have subgradients

Optimality conditions

- Subdifferential $\partial J(w) =$ (convex) set of subgradients of J at w

- From directional derivatives to subdifferential

$$g \in \partial J(w) \Leftrightarrow \forall \Delta \in \mathbb{R}^p, \; g^\top \Delta \leqslant \nabla J(w, \Delta)$$

- From subdifferential to directional derivatives

$$\nabla J(w, \Delta) = \max_{g \in \partial J(w)} g^\top \Delta$$

- Optimality conditions:
  - **Proposition**: w is optimal **if and only if** for all $\Delta \in \mathbb{R}^p$, $\nabla J(w, \Delta) \geqslant 0$
  - **Proposition**: w is optimal **if and only if** $0 \in \partial J(w)$

Subgradient and directional derivatives for ℓ_1-norm regularization

- We have with $J(w) = \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \lambda \|w\|_1$

$$\nabla J(w, \Delta) = \sum_{i=1}^{n} \ell'(y_i, w^\top x_i) x_i + \lambda \sum_{j,\ w_j \neq 0} \operatorname{sign}(w_j)^\top \Delta_j + \lambda \sum_{j,\ w_j = 0} |\Delta_j|$$

- g is a subgradient at w if and only if for all j,

$$\operatorname{sign}(w_j) \neq 0 \Rightarrow g_j = \sum_{i=1}^{n} \ell'(y_i, w^\top x_i) X_{ij} + \lambda \operatorname{sign}(w_j)$$

$$\operatorname{sign}(w_j) = 0 \Rightarrow |g_j - \sum_{i=1}^{n} \ell'(y_i, w^\top x_i) X_{ij}| \leqslant \lambda$$

Optimality conditions for ℓ_1-norm regularization

- General loss: 0 is a subgradient at w if and only if for all j,

$$\mathrm{sign}(w_j) \neq 0 \Rightarrow 0 = \sum_{i=1}^n \ell'(y_i, w^\top x_i) X_{ij} + \lambda \mathrm{sign}(w_j)$$

$$\mathrm{sign}(w_j) = 0 \Rightarrow \Big| \sum_{i=1}^n \ell'(y_i, w^\top x_i) X_{ij} \Big| \leqslant \lambda$$

- Square loss: 0 is a subgradient at w if and only if for all j,

$$\mathrm{sign}(w_j) \neq 0 \Rightarrow X(:,j)^\top (y - Xw) + \lambda \mathrm{sign}(w_j)$$

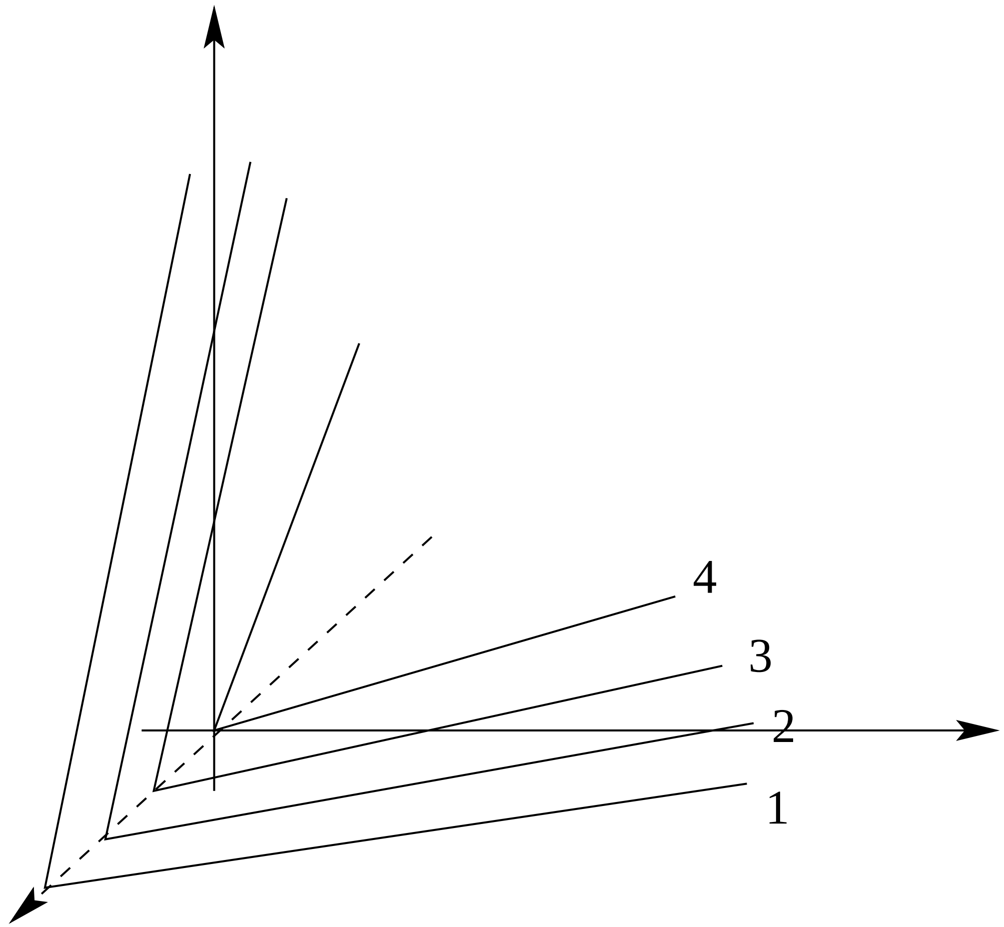
$$\mathrm{sign}(w_j) = 0 \Rightarrow |X(:,j)^\top (y - Xw)| \leqslant \lambda$$

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

- Simple case: differentiable objective
  - 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
    * diminishing step size: e.g., $\alpha_t = (t + t_0)^{-1}$
    * Linear convergence time: $O(\kappa \log(1/\varepsilon))$ iterations
  - Coordinate descent: similar properties

- Hard case: non differentiable objective
  - Subgradient descent: $w_{t+1} = w_t - \alpha_t g_t$, with $g_t \in \partial J(w_t)$
    * with exact line search: not always convergent (show counter example)
    * diminishing step size: convergent
  - Coordinate descent: not always convergent (show counterexample)

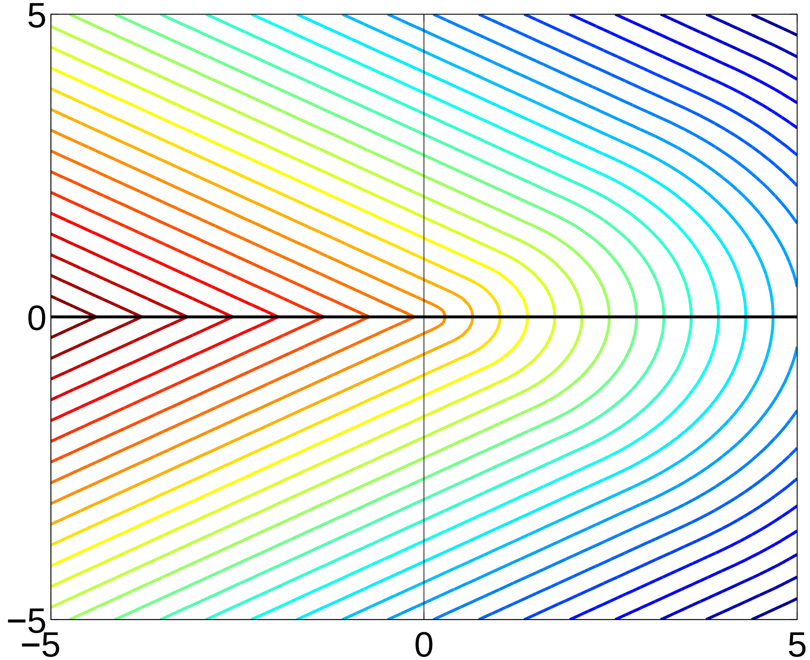
Counter-example
Coordinate descent for nonsmooth objectives



Counter-example
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 $x_1 > |x_2| > (9/16)^2|x_1|$



Second order methods

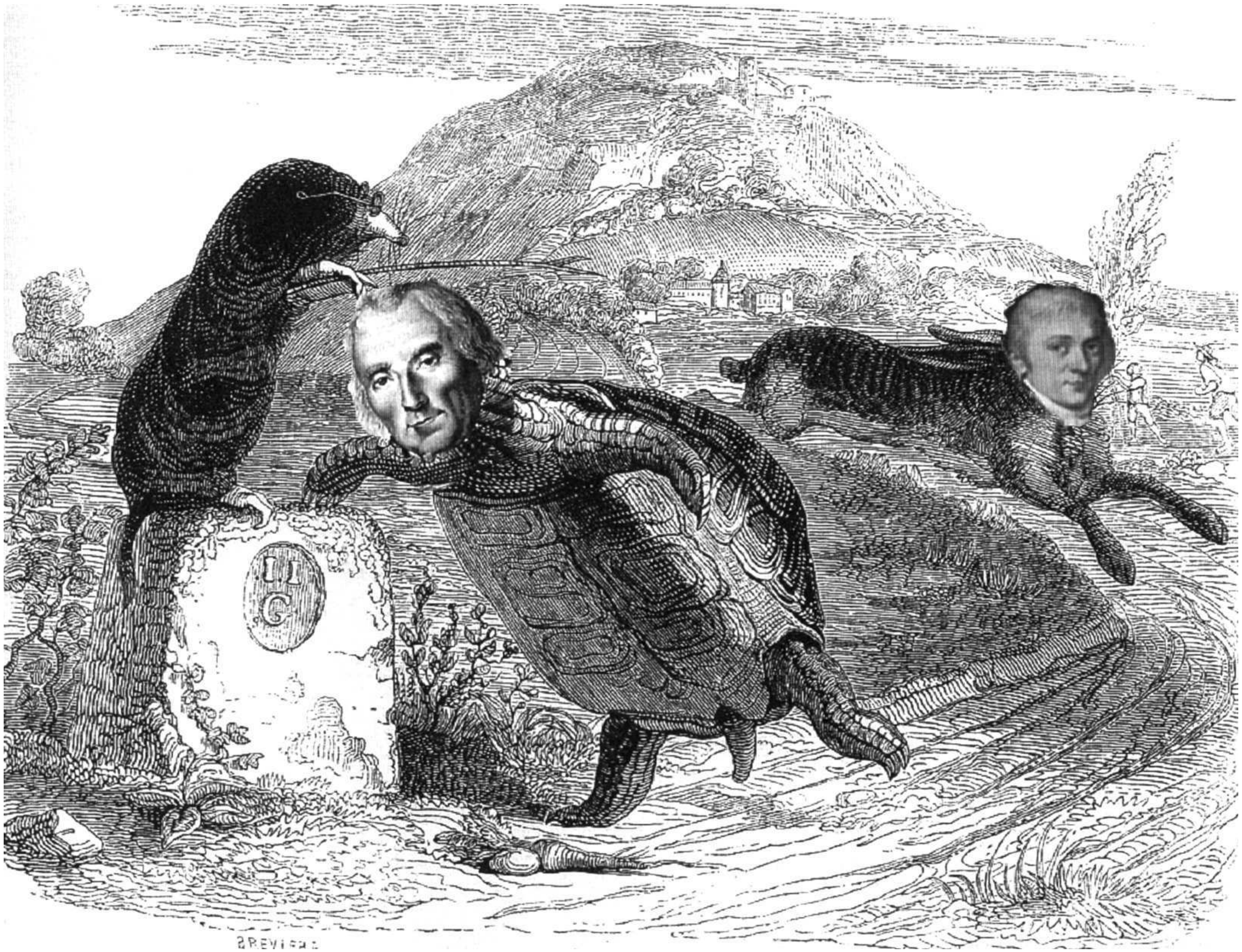
- Differentiable case
  - Newton: $w_{t+1} = w_t - \alpha_t H_t^{-1} g_t$
    * Traditional: $\alpha_t = 1$, but non globally convergent
    * globally convergent with line search for α_t (see Boyd, 2003)
    * $O(\log\log(1/\varepsilon))$ (slower) iterations
  - Quasi-newton methods (see Bonnans et 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 & Bousquet (2008) [BB08] for further details

Algorithms for ℓ^1-norms:
Gaussian hare vs. Laplacian tortoise



Cheap (and not dirty) algorithms for all losses

- Coordinate descent [WL08]
  - Globaly convergent here under reasonable assumptions!
  - very fast updates

- Subgradient descent

- Smoothing the absolute value + first/second order methods
  - Replace $|w_i|$ by $(w_i^2 + \varepsilon_i^2)^{1/2}$
  - Use gradient descent or Newton with diminishing ε

- More dedicated algorithms to get the best of both worlds: fast and precise

Special case of square loss

- Quadratic programming formulation: minimize

$$\frac{1}{2}\|y - Xw\|^2 + \lambda \sum_{j=1}^{p} (w_j^+ + w_j^-) \text{ such that } w = w^+ - w^-,\ w^+ \geqslant 0,\ w^- \geqslant 0$$

  - **generic toolboxes $\Rightarrow$ very slow**

- Main property: if the sign pattern $s \in \{-1, 0, 1\}^p$ of the solution is known, the solution can be obtained in closed form
  - Lasso equivalent to minimizing $\frac{1}{2}\|y - X_J w_J\|^2 + \lambda s_J^\top w_J$ w.r.t. w_J where $J = \{j, s_j \neq 0\}$.
  - Closed form solution $w_J = (X_J^\top X_J)^{-1}(X_J^\top Y + \lambda s_J)$

- "Simply" need to check that $\mathrm{sign}(w_J) = s_J$ and optimality for J^c

Optimality conditions for the Lasso

- 0 is a subgradient at w if and only if for all j,
  - Active variable condition

$$\text{sign}(w_j) \neq 0 \Rightarrow X(:,j)^\top (y - Xw) + \lambda \text{sign}(w_j)$$

NB: allows to compute w_J

  - Inactive variable condition

$$\text{sign}(w_j) = 0 \Rightarrow |X(:,j)^\top (y - Xw)| \leqslant \lambda$$

Algorithm 2: feature search (Lee et al., 2006, [LBRN07])

- Looking for the correct sign pattern $s \in \{-1, 0, 1\}^p$
- Initialization: start with $w = 0$, $s = 0$, $J = \{j, s_j = 0\}$
- Step 1: select $i = \arg\max_j \left|\sum_{i=1}^n \ell'(y_i, w^\top x_i) X_{ji}\right|$ and add j to the active set J with proper sign
- Step 2: find optimal vector w_{new} of $\frac{1}{2}\|y - X_J w_J\|^2 + \lambda s_J^\top w_J$
  - Perform (discrete) line search between w and w_{new}
  - Update sign of w
- Step 3: check opt. condition for active variable, if no go to step 2
- Step 4: check opt. condition for inactive variable, if no go to step 1

Algorithm 3: Lars/Lasso for the square loss [EHJT04]

- Goal: Get all solutions for all possible values of the regularization parameter λ

- Same idea as before: if the set J of active variables is known,

$$w_J^*(\lambda) = (X_J^\top X_J)^{-1}(X_J^\top Y + \lambda s_J)$$

  valid, as long as,

  – sign condition: $\mathrm{sign}(w_J^*(\lambda)) = s_J$
  – subgradient condition: $\|X_{J^c}^\top (X_J w_J^*(\lambda) - y)\|_\infty \leqslant \lambda$
  – This defines an interval on λ: the path is thus piecewise affine!

- Simply need to find break points and directions

Algorithm 3: Lars/Lasso for the square loss

- Builds a sequence of disjoint sets I_0, I_+, I_-, solutions w and parameters λ that record the break points of the path and corresponding active sets/solutions

- Initialization: $\lambda_0 = \infty$, $I_0 = \{1, \dots, p\}$, $I_+ = I_- = \varnothing$, $w = 0$

- While $\lambda_k > 0$, find minimum λ such that

$$(A) \qquad \operatorname{sign}(w_k + (\lambda - \lambda_k)(X_J^\top X_J)^{-1} s_J) = s_J$$

$$(B) \qquad \|X_{J^c}^\top (X_J w_k + (\lambda - \lambda_k) X_J (X_J^\top X_J)^{-1} s_J)\|_\infty \leqslant \lambda$$

- If (A) is blocking, remove corresponding index from I_+ or I_-

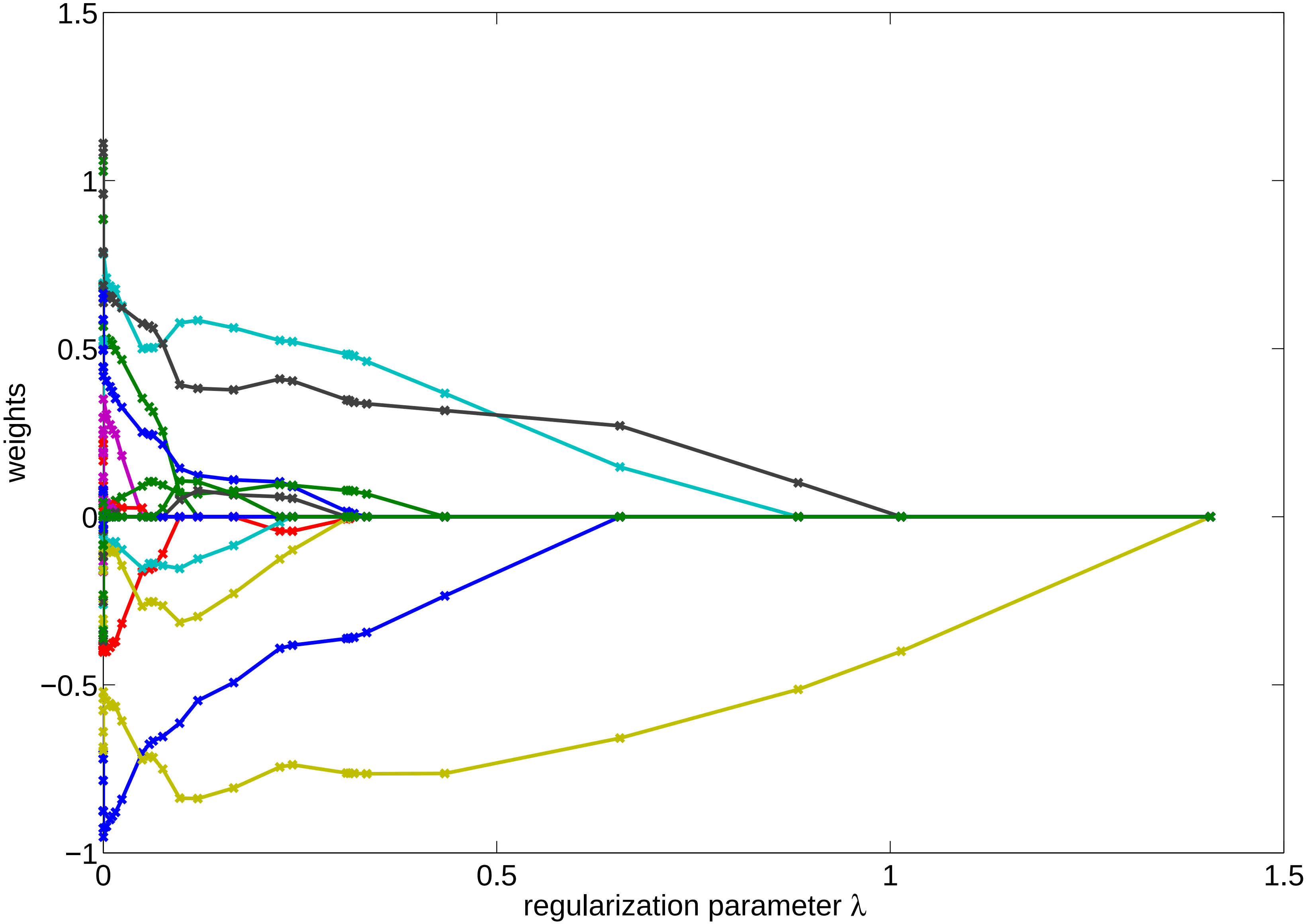
- If (B) is blocking, add corresponding index into active set I_+ or I_-

- Update corresponding λ_{k+1} and recompute w_{k+1}, $k \leftarrow k + 1$

Lasso in action

- Piecewise linear paths
- When is it supposed to work?
  - Show simulations with random Gaussians, regularization parameter estimated by cross-validation
  - sparsity is expected or not

Lasso in action

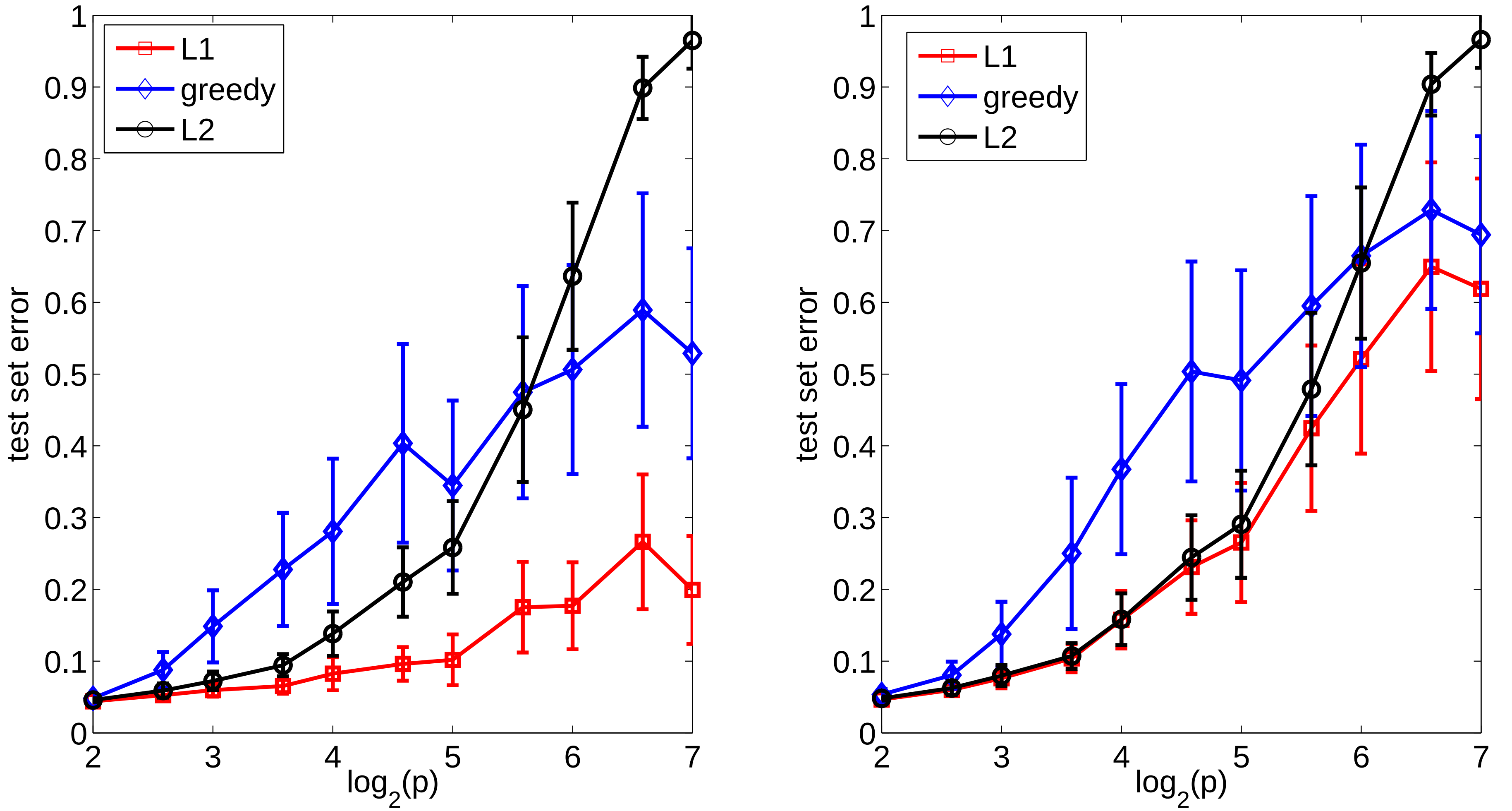


Comparing Lasso and other strategies for linear regression and subset selection

- Compared methods to reach the least-square solution [HTF01]
  - Ridge regression: $\min_w \frac{1}{2}\|y - Xw\|_2^2 + \frac{\lambda}{2}\|w\|_2^2$
  - Lasso: $\min_w \frac{1}{2}\|y - Xw\|_2^2 + \lambda\|w\|_1$
  - 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 w_{OLS}

Lasso in action



(left: sparsity is expected, right: sparsity is not expected)

ℓ^1-norm regularization and sparsity
Summary

- Nonsmooth optimization
  - subgradient, directional derivatives
  - descent methods might not always work
  - first/second order methods

- Algorithms
  - Cheap algorithms for all losses
  - Dedicated path algorithm for the square loss

Course Outline

1. ℓ^1**-norm regularization**
   - Review of nonsmooth optimization problems and algorithms
   - Algorithms for the Lasso (generic or dedicated)
   - Examples

2. **Extensions**
   - Group Lasso and multiple kernel learning (MKL) + case study
   - Sparse methods for matrices
   - Sparse PCA

3. **Theory - Consistency of pattern selection**
   - Low and high dimensional setting
   - Links with compressed sensing

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** [Aro50]: 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

Regularization and representer theorem

- Data: $x_i \in \mathbb{R}^d$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$, 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** [KW71]: 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: $\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^n \varphi_i(f^\top \Phi(x_i)) + \frac{\lambda}{2}\|f\|^2}$

	$\varphi_i(u_i)$
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

- 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):
$$f = \sum_{i=1}^n \alpha_i \Phi(x_i)$$

"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	$(1 - y_i u_i)_+$	$-vy_i \times 1_{-vy_i \in [0,1]}$

Kernel learning with convex optimization

- Kernel methods work...

...with the good kernel!

$\Rightarrow$ Why not learn the kernel directly from data?

Kernel learning with convex optimization

- Kernel methods work...

  ...with the good kernel!

  $\Rightarrow$ Why not learn the kernel directly from data?

- **Proposition** [LCG+04, BLJ04]:

$$\begin{aligned} G(K) &= \min_{f \in \mathcal{F}} \sum_{i=1}^{n} \varphi_i(f^\top \Phi(x_i)) + \frac{\lambda}{2} \|f\|^2 \\ &= \max_{\alpha \in \mathbb{R}^n} - \sum_{i=1}^{n} \psi_i(\lambda \alpha_i) - \frac{\lambda}{2} \alpha^\top K \alpha \end{aligned}$$

  is a convex function of the Gram matrix K

- Theoretical learning bounds [BLJ04]

MKL framework

- Minimize with respect to the kernel matrix K

$$G(K) = \max_{\alpha \in \mathbb{R}^n} -\sum_{i=1}^{n} \psi_i(\lambda \alpha_i) - \frac{\lambda}{2} \alpha^\top K \alpha$$

- Optimization domain:
  - K positive semi-definite in general
  - The set of kernel matrices is a cone $\rightarrow$ conic representation

$$\boxed{K(\eta) = \sum_{j=1}^{m} \eta_j K_j, \quad \eta \geqslant 0}$$

  - Trace constraints: $\operatorname{tr} K = \sum_{j=1}^{m} \eta_j \operatorname{tr} K_j = 1$

- Optimization:
  - In most cases, representation in terms of SDP, QCQP or SOCP
  - Optimization by generic toolbox is costly [BLJ04]

MKL - “reinterpretation” [BLJ04]

- Framework limited to $K = \sum_{j=1}^{m} \eta_j K_j$, $\eta \geqslant 0$
- Summing kernels is equivalent to concatenating feature spaces
  - m “feature maps” $\Phi_j : \mathcal{X} \mapsto \mathcal{F}_j$, $j = 1, \dots, m$.
  - Minimization with respect to $f_1 \in \mathcal{F}_1, \dots, f_m \in \mathcal{F}_m$
  - Predictor: $f(x) = f_1{}^\top \Phi_1(x) + \cdots + f_m{}^\top \Phi_m(x)$

$$\begin{array}{ccccc}
 & & \Phi_1(x)^\top \quad f_1 & & \\
 & \nearrow & \vdots \qquad \vdots & \searrow & \\
x & \longrightarrow & \Phi_j(x)^\top \quad f_j & \longrightarrow & f_1^\top \Phi_1(x) + \cdots + f_m^\top \Phi_m(x) \\
 & \searrow & \vdots \qquad \vdots & \nearrow & \\
 & & \Phi_m(x)^\top \quad f_m & &
\end{array}$$

  - **Which regularization?**

Regularization for multiple kernels

- Summing kernels is equivalent to concatenating feature spaces
  - m "feature maps" $\Phi_j : \mathcal{X} \mapsto \mathcal{F}_j$, $j = 1, \ldots, m$.
  - Minimization with respect to $f_1 \in \mathcal{F}_1, \ldots, f_m \in \mathcal{F}_m$
  - Predictor: $f(x) = f_1{}^\top \Phi_1(x) + \cdots + f_m{}^\top \Phi_m(x)$

- Regularization by $\sum_{j=1}^m \|f_j\|^2$ is equivalent to using $K = \sum_{j=1}^m K_j$

Regularization for multiple kernels

- Summing kernels is equivalent to concatenating feature spaces
  - m "feature maps" $\Phi_j : \mathcal{X} \mapsto \mathcal{F}_j$, $j = 1, \dots, m$.
  - Minimization with respect to $f_1 \in \mathcal{F}_1, \dots, f_m \in \mathcal{F}_m$
  - Predictor: $f(x) = f_1{}^\top \Phi_1(x) + \cdots + f_m{}^\top \Phi_m(x)$

- Regularization by $\sum_{j=1}^m \|f_j\|^2$ is equivalent to using $K = \sum_{j=1}^m K_j$

- Regularization by $\sum_{j=1}^m \|f_j\|$ should impose sparsity at the group level

- **Main questions when regularizing by block ℓ^1-norm**:

1. Equivalence with previous formulations
2. Algorithms
3. Analysis of sparsity inducing properties

MKL - duality [BLJ04]

- Primal problem:

$$\sum_{i=1}^n \varphi_i(f_1^\top \Phi_1(x_i) + \cdots + f_m^\top \Phi_m(x_i)) + \frac{\lambda}{2}\left(\|f_1\| + \cdots + \|f_m\|\right)^2$$

- **Proposition**: Dual problem (using second order cones)

$$\max_{\alpha \in \mathbb{R}^n} -\sum_{i=1}^n \psi_i(-\lambda\alpha_i) - \frac{\lambda}{2} \min_{j \in \{1,\dots,m\}} \alpha^\top K_j \alpha$$

  KKT conditions: $f_j = \eta_j \sum_{i=1}^n \alpha_i \Phi_j(x_i)$
  with $\alpha \in \mathbb{R}^n$ and $\eta \geqslant 0$, $\sum_{j=1}^m \eta_j = 1$

  – α is the dual solution for the clasical kernel learning problem with kernel matrix $K(\eta) = \sum_{j=1}^m \eta_j K_j$
  – η corresponds to the minimum of $G(K(\eta))$

Algorithms for MKL

- (very) costly optimization with SDP, QCQP ou SOCP
  - $n \geqslant 1,000 - 10,000$, $m \geqslant 100$ not possible
  - "loose" required precision $\Rightarrow$ first order methods
- Dual coordinate ascent (SMO) with smoothing [BLJ04]
- Optimization of $G(K)$ by cutting planes [SRSS06]
- Optimization of $G(K)$ with steepest descent with smoothing [RBCG08]
- Regularization path [BTJ04]

SMO for MKL [BLJ04]

- Dual function $-\sum_{i=1}^{n} \psi_i(-\lambda\alpha_i) - \frac{\lambda}{2} \min_{j \in \{1,\dots,m\}} \alpha^\top K_j \alpha$ is similar to regular SVM $\Rightarrow$ why not try SMO?

SMO for MKL

- Dual function $-\sum_{i=1}^{n} \psi_i(-\lambda\alpha_i) - \frac{\lambda}{2} \min_{j \in \{1,\dots,m\}} \alpha^\top K_j \alpha$ is similar to regular SVM $\Rightarrow$ why not try SMO?

  – Non differentiability!

SMO for MKL

- Dual function $-\sum_{i=1}^{n} \psi_i(-\lambda\alpha_i) - \frac{\lambda}{2} \min_{j\in\{1,\dots,m\}} \alpha^\top K_j \alpha$ is similar to regular SVM $\Rightarrow$ why not try SMO?

  – Non differentiability!
  – Solution: smoothing of the dual function by adding a squared norm in the primal problem (Moreau-Yosida regularization)

$$\min_f \sum_{i=1}^{n} \varphi_i\Big(\sum_{j=1}^{m} f_j^\top \Phi_j(x_i)\Big) + \frac{\lambda}{2}\left(\sum_{j=1}^{m} \|f_j\|\right)^2 + \varepsilon \sum_{j=1}^{m} \|f_j\|^2$$

- SMO for MKL: simply descent on the dual function

- Matlab/C code available online (Obozinsky, 2006)

Could we use previous implementations of SVM?

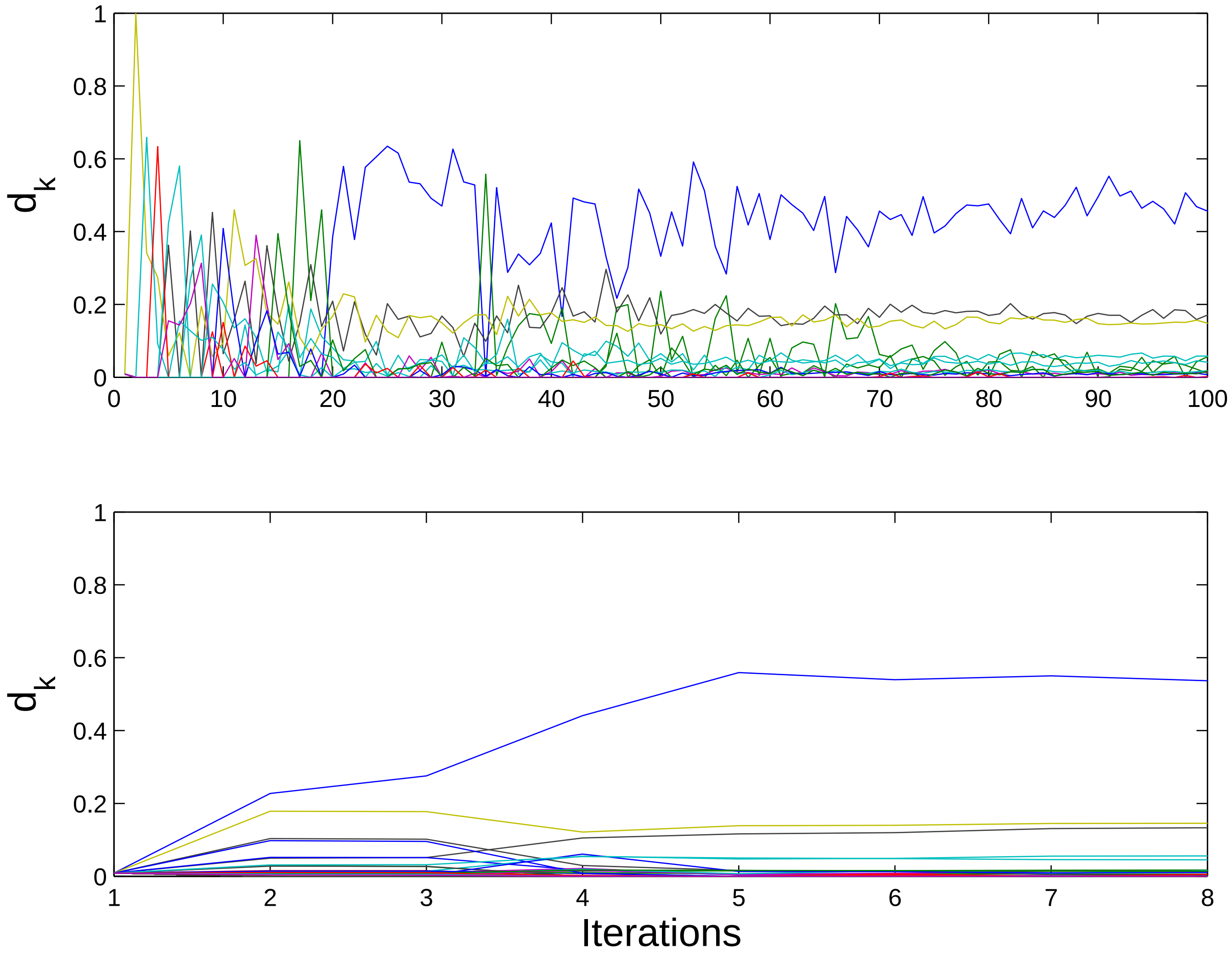
- Computing one value and one subgradient of

$$G(\eta) = \max_{\alpha \in \mathbb{R}^n} -\sum_{i=1}^n \psi_i(\lambda\alpha_i) - \frac{\lambda}{2}\alpha^\top K(\eta)\alpha$$

requires to solve a classical problem (e.g., SVM)

- Optimization of η directly
  - Cutting planes [SRSS06]
  - Gradient descent [RBCG08]

Direct optimization of $G(\eta)$ [RBCG08]



MKL with regularization paths [BTJ04]

- Regularized problen

$$\sum_{i=1}^{n} \phi_i(w_1^\top \Phi_1(x_i) + \cdots + w_m^\top \Phi_m(x_i)) + \frac{\lambda}{2}(\|w_1\| + \cdots + \|w_m\|)^2$$

- In practice, solution required for "many" parameters λ

- Can we get all solutions at the cost of one?
  - Rank one kernels (usual ℓ_1 norm): path is piecewise affine for some losses $\Rightarrow$ Exact methods [EHJT04, HRTZ05, BHH06]
  - Rank > 1: path is only est piecewise smooth
    $\Rightarrow$ predictor-corrector methods [BTJ04]

Log-barrier regularization

- Dual problem:

$$\max_\alpha - \sum_i \psi_i(\lambda \alpha_i) \text{ such that } \forall j, \alpha^\top K_j \alpha \leqslant d_j^2$$

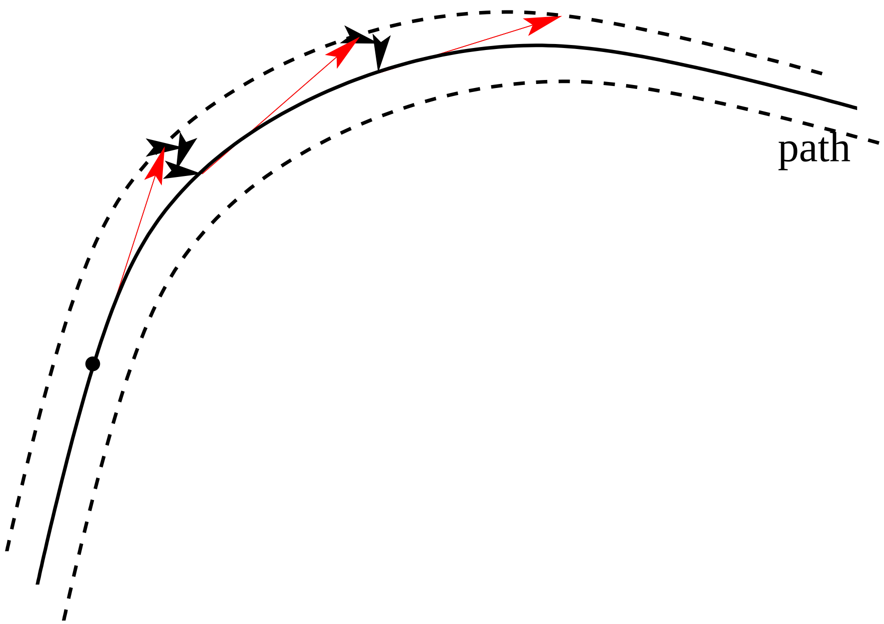
- Regularized dual problem:

$$\max_\alpha - \sum_i \psi_i(\lambda \alpha_i) + \mu \sum_j \log(d_j^2 - \alpha^\top K_j \alpha)$$

- Properties:
  - Unconstrained concave maximization
  - η function of α
  - α is unique solution of the stationary equation $F(\alpha, \lambda) = 0$
  - $\alpha(\lambda)$ differentiable function, easy to follow

Predictor-corrector method

- Follow solution of $F(\alpha, \lambda) = 0$
- Predictor steps
  - First order approximation using $\frac{d\alpha}{d\lambda} = -\left(\frac{\partial F}{\partial \alpha}\right)^{-1} \frac{\partial F}{\partial \lambda}$
- Corrector steps
  - Newton's method to converge back to solution



Link with interior point methods

- Regularized dual problem:

$$\max_\alpha -\sum_i \psi_i(\lambda\alpha_i) + \mu \sum_j \log(d_j^2 - \alpha^\top K_j \alpha)$$

- Interior point methods:
  - λ fixed, μ followed from large to small
- Regularization path:
  - μ fixed small, λ followed from large to small
- Computational complexity: Total complexity $O(mn^3)$
  - NB: sparsity in α not used

Applications

- Bioinformatics [LBC+04]
  - Protein function prediction
  - Heterogeneous data sources
    - Amino acid sequences
    - Protein-protein interactions
    - Genetic interactions
    - Gene expression measurements
- Image annotation [HB07]

A case study in kernel methods

- Goal: show how to use kernel methods (kernel design + kernel learning) on a "real problem"

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

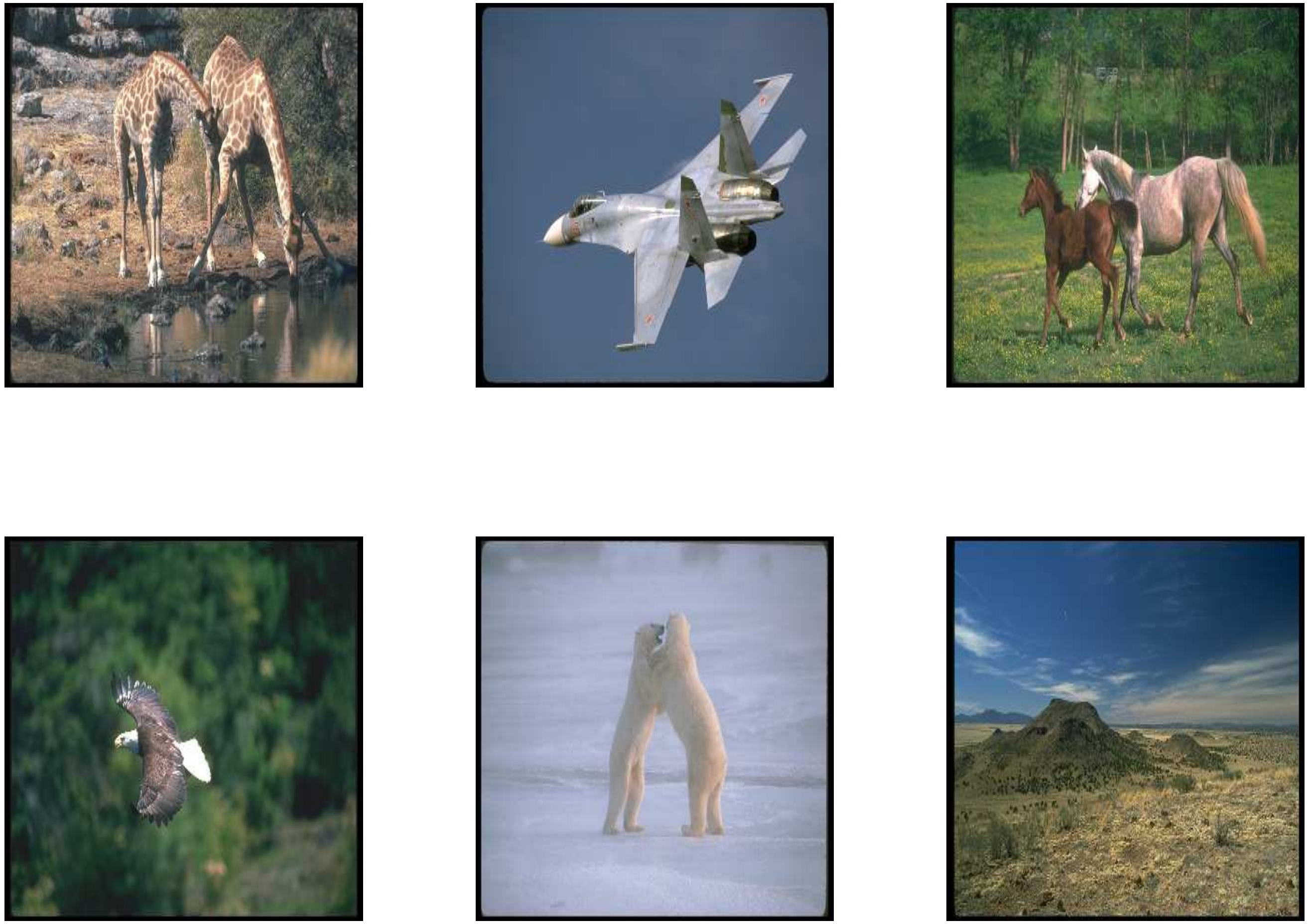
Kernel trick and modularity

- 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

Image annotation and kernel design

- `Corel14`: 1400 *natural images* with 14 classes



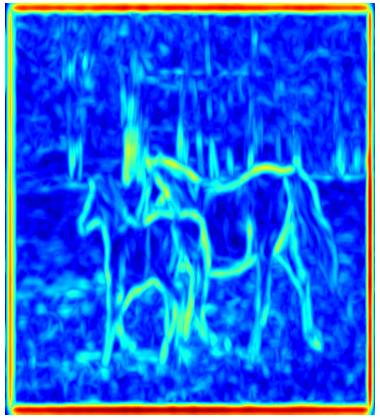
Segmentation

- Goal: 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)

Segmentation with the watershed transform

image

gradient

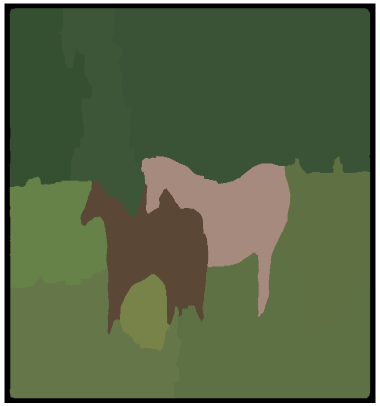


watershed

287 segments

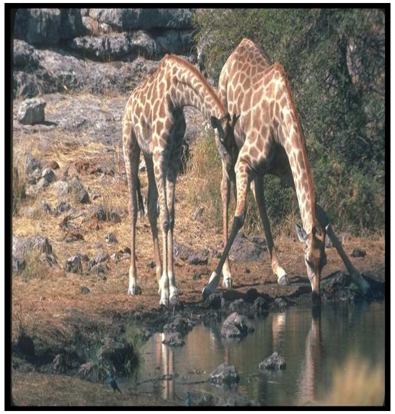
64 segments

10 segments

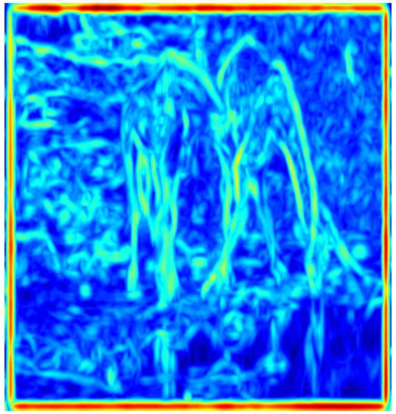


Segmentation with the watershed transform

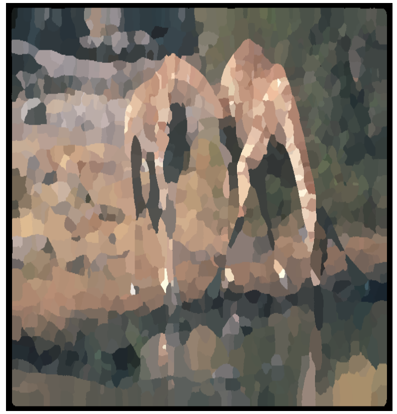
image



gradient



watershed



287 segments

64 segments

10 segments

Image as a segmentation graph

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

$\Rightarrow$

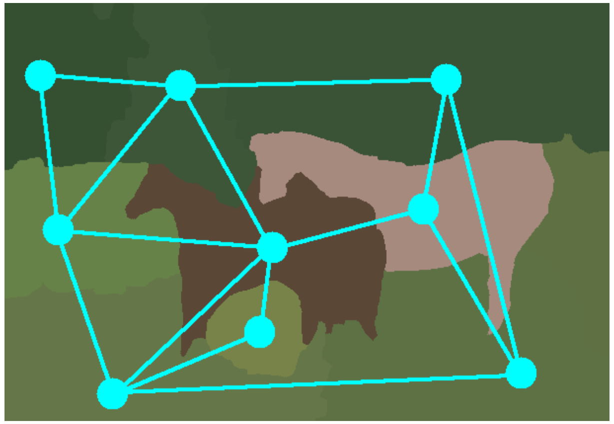


Image as a segmentation graph

- 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

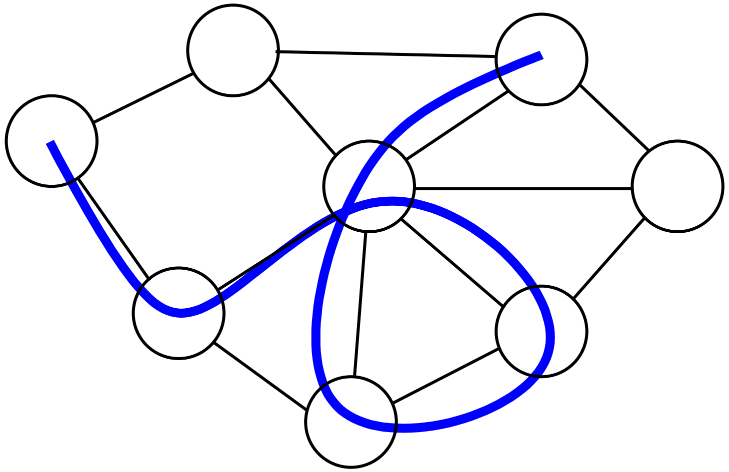
- **Graph kernels** [GFW03] provide an elegant and efficient solution

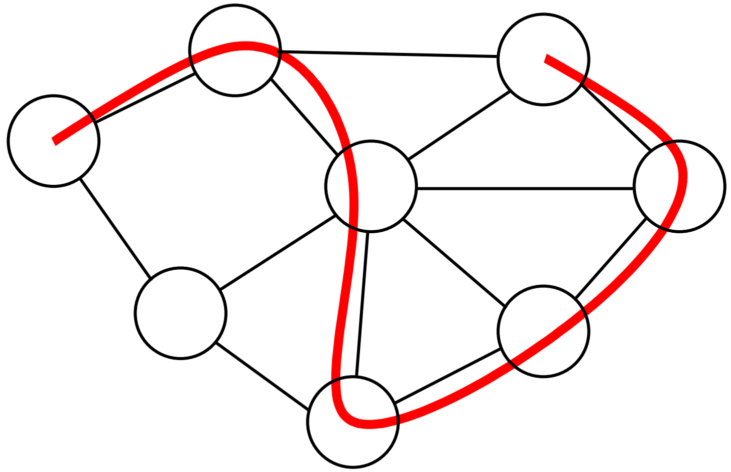
Kernels between structured objects Strings, graphs, etc... [STC04]

- Numerous applications (text, bio-informatics)
- From probabilistic models on objects (e.g., Saunders et al, 2003)
- Enumeration of subparts (Haussler, 1998, Watkins, 1998)
  - Efficient for strings
  - Possibility of gaps, partial matches, very efficient algorithms (Leslie et al, 2002, Lodhi et al, 2002, etc...)
- Most approaches fails for general graphs (even for undirected trees!)
  - NP-Hardness results (Gärtner et al, 2003)
  - Need alternative set of subparts

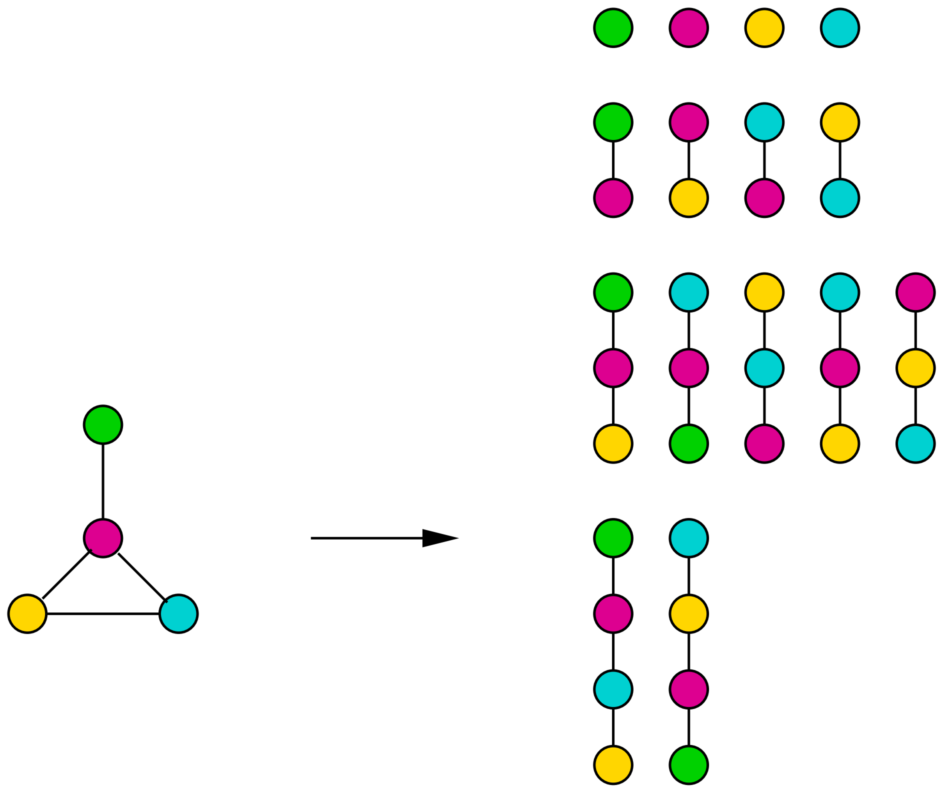
Paths and walks

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

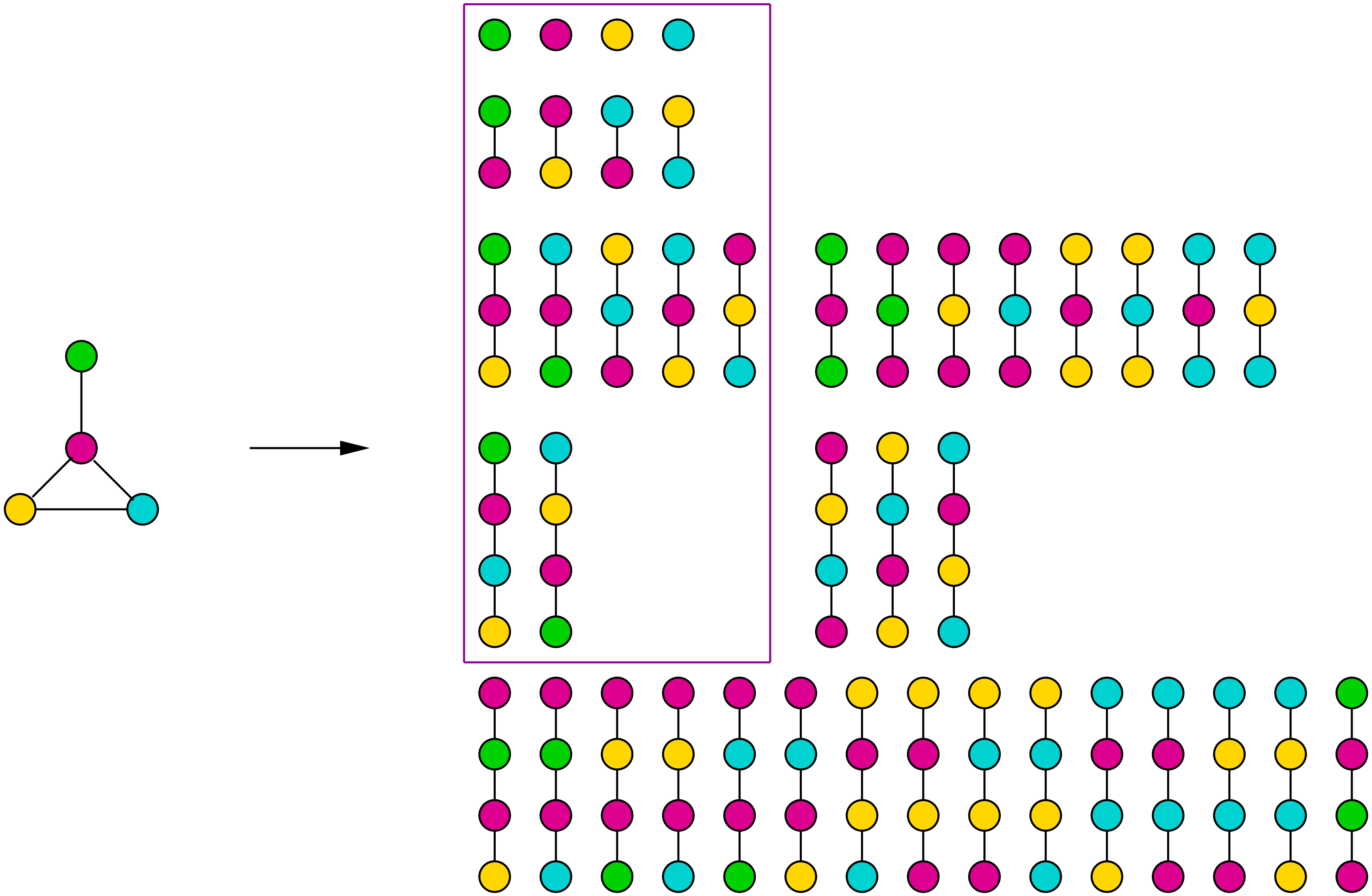




Paths



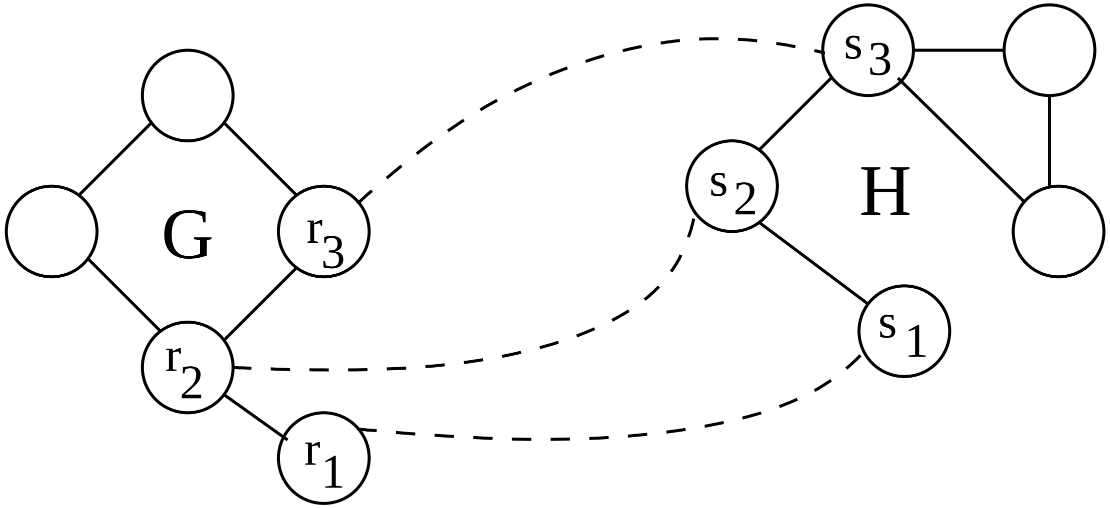
Walks



Walk kernel (Kashima, 2004, Borgwardt, 2005)

- $\mathcal{W}_{\mathbf{G}}^p$ (resp. $\mathcal{W}_{\mathbf{H}}^p$) 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:

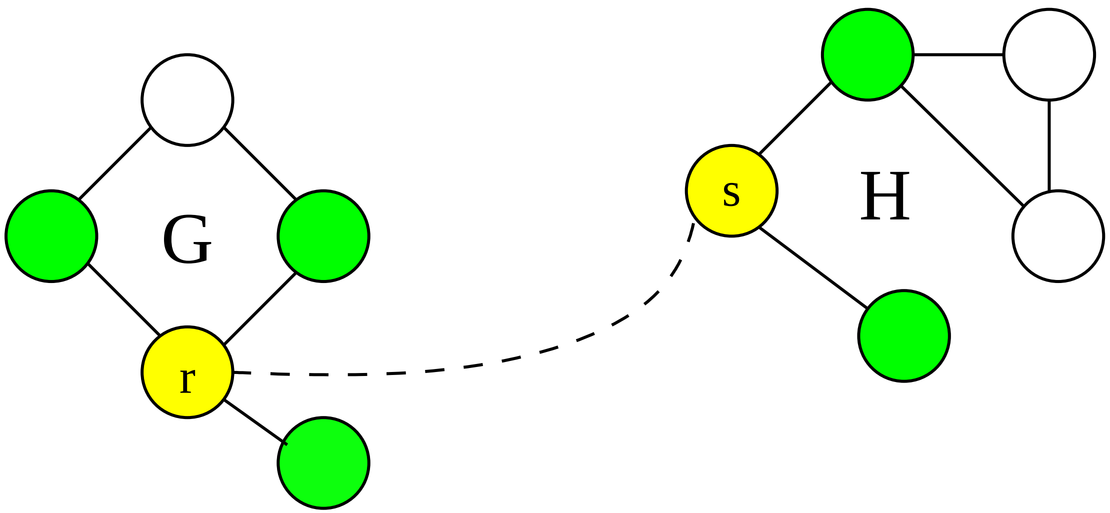
$$k_{\mathcal{W}}^p(\mathbf{G}, \mathbf{H}) = \sum_{\substack{(r_1, \ldots, r_p) \in \mathcal{W}_{\mathbf{G}}^p \\ (s_1, \ldots, s_p) \in \mathcal{W}_{\mathbf{H}}^p}} \prod_{i=1}^p k(\ell_{\mathbf{G}}(r_i), \ell_{\mathbf{H}}(s_i)).$$



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

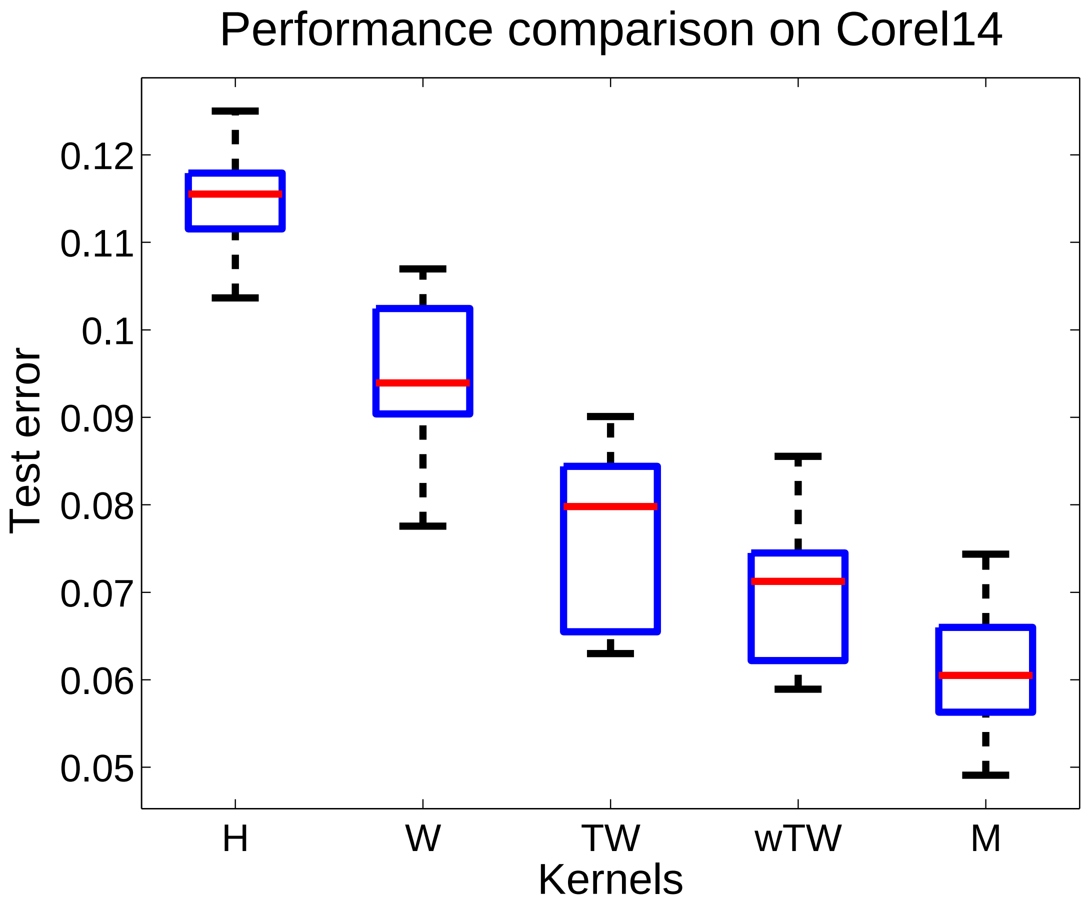
- recursion between $p-1$-th walk and p-th walk kernel

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

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

Performance on Corel14
(Harchaoui & Bach, 2007)

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



MKL
Summary

- Block ℓ^1-norm extends regular ℓ^1-norm
- One kernel per block
- Application:
  - Data fusion
  - Hyperparameter selection
  - Non linear variable selection

Course Outline

1. **ℓ^1-norm regularization**
   - Review of nonsmooth optimization problems and algorithms
   - Algorithms for the Lasso (generic or dedicated)
   - Examples

2. **Extensions**
   - Group Lasso and multiple kernel learning (MKL) + case study
   - Sparse methods for matrices
   - Sparse PCA

3. **Theory - Consistency of pattern selection**
   - Low and high dimensional setting
   - Links with compressed sensing

Learning on matrices

- Example 1: matrix completion
  - Given a matrix $M \in \mathbb{R}^{n \times p}$ and a subset of observed entries, estimate all entries
  - Many applications: graph learning, collaborative filtering [BHK98, HCM$^+$00, SMH07]

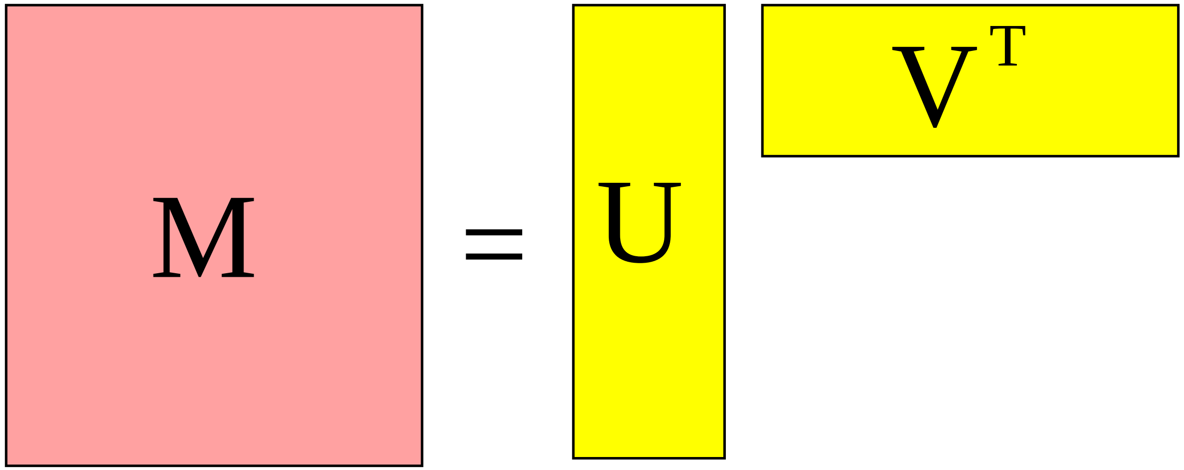
- Example 2: multi-task learning [OTJ07, PAE07]
  - Common features for m learning problems $\Rightarrow$ m different weights, i.e., $W = (w_1, \ldots, w_m) \in \mathbb{R}^{p \times m}$
  - Numerous applications

- Example 3: image denoising [EA06, MSE08]
  - Simultaneously denoise all patches of a given image

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

1. A lot of zero elements
   - does not use the matrix structure!

2. A small rank
   - $M = UV^\top$ where $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{n \times m}$, m small
   - **Trace norm**



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

1. A lot of zero elements
   - does not use the matrix structure!

2. A small rank
   - $M = UV^\top$ where $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{n \times m}$, m small
   - **Trace norm**

3. A decomposition into sparse (but large) matrix $\Rightarrow$ redundant dictionaries
   - $M = UV^\top$ where $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{n \times m}$, U sparse
   - **Dictionary learning**

Trace norm [SRJ05, FHB01, Bac08c]

- Singular value decomposition: $M \in \mathbb{R}^{n \times p}$ can always be decomposed into $M = U \operatorname{Diag}(s) V^\top$, where $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{n \times m}$ have orthonormal columns and s is a positive vector (of singular values)

- ℓ^0 norm of singular values $=$ rank

- ℓ^1 norm of singular values $=$ trace norm

- Similar properties than the ℓ^1-norm
  - Convexity
  - Solutions of penalized problem have low rank
  - Algorithms

Dictionary learning [EA06, MSE08]

- Given $X \in \mathbb{R}^{n \times p}$, i.e., n vectors in $\mathbb{R}^p$, find
  - m dictionary elements in $\mathbb{R}^p$: $V = (v_1, \dots, v_m) \in \mathbb{R}^{p \times m}$
  - m set of decomposition coefficients: $U =\in \mathbb{R}^{n \times m}$
  - such that U is sparse and small reconstruction error, i.e., $\|X - UV^\top\|_F^2 = \sum_{i=1}^n \|X(i,:) - U(i,:)V^\top\|_2^2$ is small

- NB: Opposite view: not sparse in term of ranks, sparse in terms of decomposition coefficients

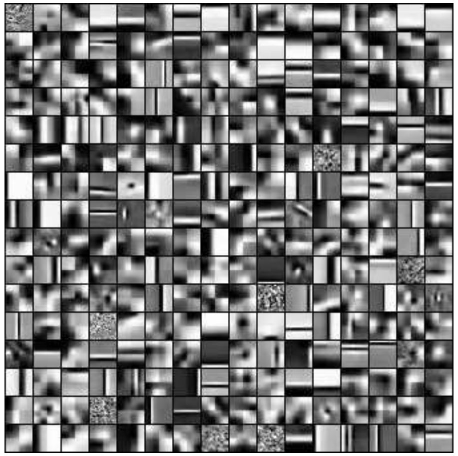
- Minimize with respect to U and V, such that $\|V(:,i)\|_2 = 1$,

$$\frac{1}{2}\|X - UV^\top\|_F^2 + \lambda \sum_{i=1}^N \|U(i,:)\|_1$$

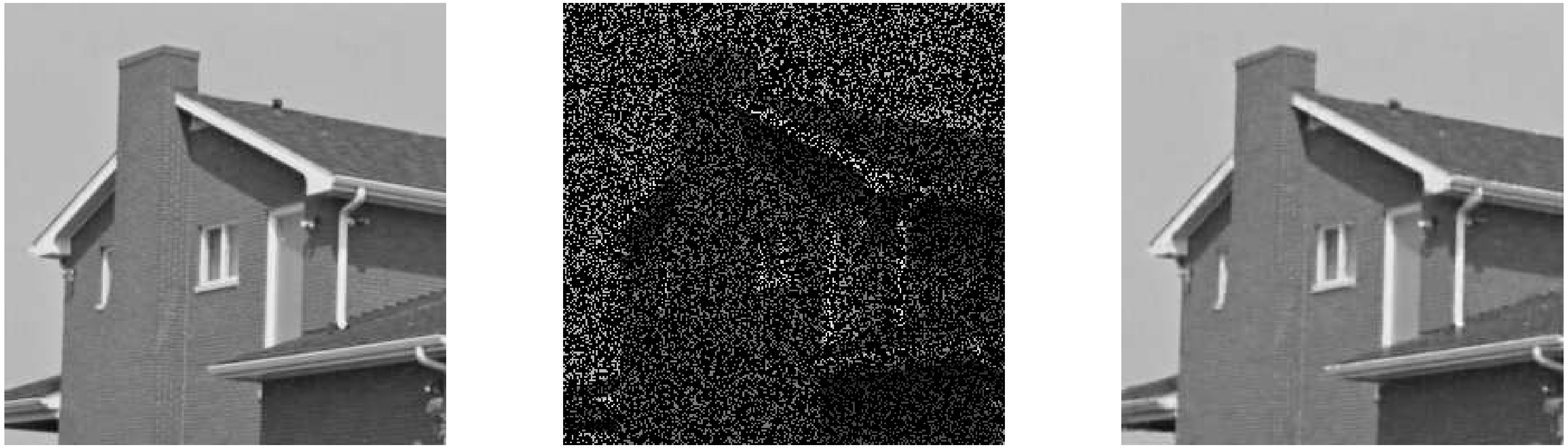
  - non convex, alternate minimization

Dictionary learning - Applications [MSE08]

- Applications in image denoising



Dictionary learning - Applications - Inpainting



Sparse PCA [DGJL07, ZHT06]

- Consider $\Sigma = \frac{1}{n} X^\top X \in \mathbb{R}^{p \times p}$ covariance matrix
- Goal: find a unit norm vector x with maximum variance $x^\top \Sigma x$ and minimum cardinality
- Combinatorial optimization problem: $\displaystyle \max_{\|x\|_2=1} x^\top \Sigma x + \rho \|x\|_0$
- First relaxation: $\|x\|_2 = 1 \Rightarrow \|x\|_1 \leqslant \|x\|_0^{1/2}$
- Rewriting using $X = xx^\top$: $\|x\|_2 = 1 \Leftrightarrow \operatorname{tr} X = 1$, $1^\top |X| 1 = \|x\|_1^2$

$$\max_{X \succcurlyeq 0,\ \operatorname{tr} X = 1,\ \operatorname{rank}(X)=1} \operatorname{tr} X\Sigma + \rho 1^\top |X| 1$$

Sparse PCA [DGJL07, ZHT06]

- Sparse PCA problem equivalent to

$$\max_{X \succcurlyeq 0, \ \operatorname{tr} X = 1, \ \operatorname{rank}(X) = 1} \operatorname{tr} X\Sigma + \rho 1^\top |X| 1$$

- Convex relaxation: dropping the rank constraint $\operatorname{rank}(X) = 1$

$$\boxed{\max_{X \succcurlyeq 0, \operatorname{tr} X = 1} \operatorname{tr} X\Sigma + \rho 1^\top |X| 1}$$

- Semidefinite program [BV03]
- Deflation to get multiple components
- "dual problem" to dictionary learning

Sparse PCA [DGJL07, ZHT06]

- Non-convex formulation

$$\min_{\alpha^\top \alpha = I} \|(I - \alpha\beta^\top)X\|_F^2 + \lambda\|\beta\|_1$$

- Dual to sparse dictionary learning

Sparse ???

-

Summary

- Notion of sparsity quite general
- Interesting links with convexity
  - Convex relaxation
- Sparsifying the world
  - All linear methods can be kernelized
  - All linear methods can be sparsified
    - Sparse PCA
    - Sparse LDA
    - Sparse ....

Course Outline

1. ℓ^1-**norm regularization**
   - Review of nonsmooth optimization problems and algorithms
   - Algorithms for the Lasso (generic or dedicated)
   - Examples

2. **Extensions**
   - Group Lasso and multiple kernel learning (MKL) + case study
   - Sparse methods for matrices
   - Sparse PCA

3. **Theory - Consistency of pattern selection**
   - Low and high dimensional setting
   - Links with compressed sensing

Theory

- Sparsity-inducing norms often used heuristically
- When does it converge to the correct pattern?
  - Yes if certain conditions on the problem are satisfied (low correlation)
  - what if not?
- Links with compressed sensing

Model consistency of the Lasso

- Sparsity-inducing norms often used heuristically
- If the responses $y_1, \ldots, y_n$ are such that $y_i = w_0^\top x_i + \varepsilon_i$ where ε_i are i.i.d. and w_0 is sparse, do we get back the correct pattern of zeros?
- Intuitive answer: yes **if and ony if** some consistency condition on the generating covariance matrices is satisfied [ZY06, YL07, Zou06, Wai06]

Asymptotic analysis - Low dimensional setting

- Asymptotic set up
  - data generated from linear model $Y = X^\top \mathbf{w} + \varepsilon$
  - $\hat{w}$ any minimizer of the Lasso problem
  - number of observations n tends to infinity

- Three types of consistency
  - regular consistency: $\|\hat{w} - \mathbf{w}\|_2$ tends to zero in probability
  - pattern consistency: the sparsity pattern $\hat{J} = \{j, \ \hat{w}_j \neq 0\}$ tends to $\mathbf{J} = \{j, \ \mathbf{w}_j \neq 0\}$ in probability
  - sign consistency: the sign vector $\hat{s} = \mathrm{sign}(\hat{w})$ tends to $\mathbf{s} = \mathrm{sign}(\mathbf{w})$ in probability

- NB: with our assumptions, pattern and sign consistencies are equivalent once we have regular consistency

Assumptions for analysis

- Simplest assumptions (fixed p, large n):
  1. Sparse linear model: $Y = X^\top \mathbf{w} + \varepsilon$, ε independent from X, and $\mathbf{w}$ sparse.
  2. Finite cumulant generating functions $\mathbb{E}\exp(a\|X\|_2^2)$ and $\mathbb{E}\exp(a\varepsilon^2)$ finite for some $a > 0$ (e.g., Gaussian noise)
  3. Invertible matrix of second order moments $\mathbf{Q} = \mathbb{E}(XX^\top) \in \mathbb{R}^{p\times p}$.

Asymptotic analysis - simple cases

$$\min_{w \in \mathbb{R}^p} \frac{1}{2n} \|Y - Xw\|_2^2 + \mu_n \|w\|_1$$

- **If μ_n tends to infinity**
  - $\hat{w}$ tends to zero with probability tending to one
  - $\hat{J}$ tends to $\varnothing$ in probability

Asymptotic analysis - simple cases

$$\min_{w\in\mathbb{R}^p} \frac{1}{2n}\|y - Xw\|_2^2 + \mu_n\|w\|_1$$

- **If μ_n tends to infinity**
  - $\hat{w}$ tends to zero with probability tending to one
  - $\hat{J}$ tends to $\varnothing$ in probability
- **If μ_n tends to $\mu_0 \in (0, \infty)$**
  - $\hat{w}$ converges to the minimum of $\frac{1}{2}(w - \mathbf{w})^\top \mathbf{Q}(w - \mathbf{w}) + \mu_0\|w\|_1$
  - The sparsity and sign patterns may or may not be consistent
  - Possible to have sign consistency without regular consistency

Asymptotic analysis - simple cases

$$\min_{w \in \mathbb{R}^p} \frac{1}{2n} \|Y - Xw\|_2^2 + \mu_n \|w\|_1$$

- **If μ_n tends to infinity**
  - $\hat{w}$ tends to zero with probability tending to one
  - $\hat{J}$ tends to $\varnothing$ in probability
- **If μ_n tends to $\mu_0 \in (0, \infty)$**
  - $\hat{w}$ converges to the minimum of $\frac{1}{2}(w - \mathbf{w})^\top \mathbf{Q}(w - \mathbf{w}) + \mu_0 \|w\|_1$
  - The sparsity and sign patterns may or may not be consistent
  - Possible to have sign consistency without regular consistency
- **If μ_n tends to zero faster than $n^{-1/2}$**
  - $\hat{w}$ converges in probability to $\mathbf{w}$
  - With probability tending to one, all variables are included

Asymptotic analysis - important case

$$\min_{w\in\mathbb{R}^p} \frac{1}{2n}\|Y - Xw\|_2^2 + \mu_n\|w\|_1$$

- **If μ_n tends to zero slower than $n^{-1/2}$**
  - $\hat{w}$ converges in probability to $\mathbf{w}$
  - the sign pattern converges to the one of the minimum of

$$\frac{1}{2}v^\top \mathbf{Q} v + v_{\mathbf{J}}^\top \mathrm{sign}(\mathbf{w}_{\mathbf{J}}) + \|v_{\mathbf{J}^c}\|_1$$

  - The sign pattern is equal to $\mathbf{s}$ (i.e., sign consistency) if and only if

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

  - Consistency condition found by many authors: Yuan & Lin (2007), Wainwright (2006), Zhao & Yu (2007), Zou (2006)

Proof (μ_n tends to zero slower than $n^{-1/2}$) - I

- Write $y = X\mathbf{w} + \varepsilon$

$$\begin{aligned}
\frac{1}{n}\|y - Xw\|_2^2 &= \frac{1}{n}\|X(\mathbf{w} - w) + \varepsilon\|_2^2 \\
&= (\mathbf{w} - w)^\top \left(\frac{1}{n}X^\top X\right)(\mathbf{w} - w) + \frac{1}{n}\|\varepsilon\|_2^2 + \frac{2}{n}(\mathbf{w} - w)^\top X^\top \varepsilon
\end{aligned}$$

- Write $w = \mathbf{w} + \mu_n \Delta$. Cost function (up to constants):

$$\begin{aligned}
&\frac{1}{2}\mu_n^2 \Delta^\top \left(\frac{1}{n}X^\top X\right)\Delta - \frac{1}{n}\mu_n \Delta^\top X^\top \varepsilon + \mu_n \left(\|\mathbf{w} + \mu_n \Delta\|_1 - \|\mathbf{w}\|_1\right) \\
=&\frac{1}{2}\mu_n^2 \Delta^\top \left(\frac{1}{n}X^\top X\right)\Delta - \frac{1}{n}\mu_n \Delta^\top X^\top \varepsilon + \mu_n \left(\mu_n \|\Delta_{\mathbf{J}^c}\|_1 + \mu_n \mathrm{sign}(\mathbf{w}_\mathbf{J})^\top \Delta_\mathbf{J}\right)
\end{aligned}$$

Proof (μ_n tends to zero slower than $n^{-1/2}$) - II

- Write $w = \mathbf{w} + \mu_n \Delta$. Cost function (up to constants):

$$\frac{1}{2}\mu_n^2 \Delta^\top \left(\frac{1}{n} X^\top X\right) \Delta - \frac{1}{n}\mu_n \Delta^\top X^\top \varepsilon + \mu_n \left(\|\mathbf{w} + \mu_n \Delta\|_1 - \|\mathbf{w}\|_1\right)$$

$$= \frac{1}{2}\mu_n^2 \Delta^\top \left(\frac{1}{n} X^\top X\right) \Delta - \frac{1}{n}\mu_n \Delta^\top X^\top \varepsilon + \mu_n \left(\mu_n \|\Delta_{\mathbf{J}^c}\|_1 + \mu_n \mathrm{sign}(\mathbf{w}_\mathbf{J})^\top \Delta_\mathbf{J}\right)$$

- Asymptotics 1: $\frac{1}{n} X^\top \varepsilon = O_p(n^{-1/2})$ negligible compared to μ_n (TCL)
- Asymptotics 2: $\frac{1}{n} X^\top X$ "converges" to $\mathbf{Q}$ (covariance matrix)
- Δ is thus the minimum of $\frac{1}{2}\Delta^\top \mathbf{Q} \Delta + \Delta_\mathbf{J}^\top \mathrm{sign}(\mathbf{w}_\mathbf{J}) + \|\Delta_{\mathbf{J}^c}\|_1$
- Check when the previous problem has solution such that $\Delta_{\mathbf{J}^c} = 0$

Proof (μ_n tends to zero slower than $n^{-1/2}$) - II

- Write $w = \mathbf{w} + \mu_n \Delta$.
- Asymptotics $\Rightarrow$ Δ minimum of $\frac{1}{2}\Delta^\top \mathbf{Q} \Delta + \Delta_{\mathbf{J}}^\top \mathrm{sign}(\mathbf{w}_{\mathbf{J}}) + \|\Delta_{\mathbf{J}^c}\|_1$
- Check when the previous problem has solution such that $\Delta_{\mathbf{J}^c} = 0$
- Solving for $\Delta_{\mathbf{J}}$: $\Delta_{\mathbf{J}} = -\mathbf{Q}_{\mathbf{J}\mathbf{J}}^{-1} \mathrm{sign}(\mathbf{w}_{\mathbf{J}})$
- Subgradient:
  - on variables in $\mathbf{J}$: equal to zero
  - on variables in $\mathbf{J}^c$: $\mathbf{Q}_{\mathbf{J}^c\mathbf{J}} \Delta_{\mathbf{J}} + g$ such that $\|g\|_\infty \leqslant 1$
- Optimality conditions: $\|\mathbf{Q}_{\mathbf{J}^c\mathbf{J}} \mathbf{Q}_{\mathbf{J}\mathbf{J}}^{-1} \mathrm{sign}(\mathbf{w}_{\mathbf{J}})\|_\infty \leqslant 1$

Asymptotic analysis

$$\min_{w \in \mathbb{R}^p} \frac{1}{2n} \|Y - Xw\|_2^2 + \mu_n \|w\|_1$$

- **If μ_n tends to zero slower than $n^{-1/2}$**
  - $\hat{w}$ converges in probability to $\mathbf{w}$
  - the sign pattern converges to the one of the minimum of

$$\frac{1}{2} v^\top \mathbf{Q} v + v_{\mathbf{J}}^\top \operatorname{sign}(\mathbf{w}_{\mathbf{J}}) + \|v_{\mathbf{J}^c}\|_1$$

  - The sign pattern is equal to $\mathbf{s}$ (i.e., sign consistency) if and only if

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

  - Consistency condition found by many authors: Yuan & Lin (2007), Wainwright (2006), Zhao & Yu (2007), Zou (2006)
  - Disappointing?

Summary of asymptotic analysis

$\lim \mu_n$ $\lim n^{1/2}\mu_n$	$+\infty$ $+\infty$	$\mu_0 \in (0,\infty)$ $+\infty$	0 $+\infty$	0 $\nu_0 \in (0,\infty)$	0 0
regular consistency	inconsistent	inconsistent	consistent	consistent	consistent
sign pattern	no variable selected	deterministic pattern (depending on μ_0)	deterministic pattern	??	all variables selected

- **If μ_n tends to zero exactly at rate $n^{-1/2}$?**

Summary of asymptotic analysis

$\lim \mu_n$ $\lim n^{1/2}\mu_n$	$+\infty$ $+\infty$	$\mu_0 \in (0,\infty)$ $+\infty$	0 $+\infty$	0 $\nu_0 \in (0,\infty)$	0 0
regular consistency	inconsistent	inconsistent	consistent	consistent	consistent
sign pattern	no variable selected	deterministic pattern (depending on μ_0)	deterministic pattern	all patterns consistent on $\mathbf{J}$, with proba. > 0	all variables selected

- **If μ_n tends to zero exactly at rate $n^{-1/2}$?**

Positive or negative result?

- Rather negative: Lasso does not always work!

- Making the Lasso consistent
  - Adaptive Lasso: reweight the ℓ^1 using ordinary least-square estimate, i.e., replace $\sum_{i=1}^p |w_i|$ by $\sum_{i=1}^p \frac{|w_i|}{|\hat{w}_i^{OLS}|}$
    $\Rightarrow$ provable consistency in all cases
  - Using the bootstrap $\Rightarrow$ Bolasso [Bac08a]

Asymptotic analysis

- **If μ_n tends to zero at rate $n^{-1/2}$, i.e., $n^{1/2}\mu_n \to \nu_0 \in (0, \infty)$**
  - $\hat{w}$ converges in probability to $\mathbf{w}$
  - All (and only) patterns which are consistent with $\mathbf{w}$ on $\mathbf{J}$ are attained with positive probability

Asymptotic analysis

- **If μ_n tends to zero at rate $n^{-1/2}$, i.e., $n^{1/2}\mu_n \to \nu_0 \in (0,\infty)$**
  - $\hat{w}$ converges in probability to $\mathbf{w}$
  - All (and only) patterns which are consistent with $\mathbf{w}$ on $\mathbf{J}$ are attained with positive probability
  - **Proposition**: for any pattern $s \in \{-1,0,1\}^p$ such that $s_{\mathbf{J}} \neq \operatorname{sign}(\mathbf{w}_{\mathbf{J}})$, there exist a constant $A(\mu_0) > 0$ such that

$$\log \mathbb{P}(\operatorname{sign}(\hat{w}) = s) \leqslant -nA(\mu_0) + O(n^{-1/2}).$$

  - **Proposition**: for any sign pattern $s \in \{-1,0,1\}^p$ such that $s_{\mathbf{J}} = \operatorname{sign}(\mathbf{w}_{\mathbf{J}})$, $\mathbb{P}(\operatorname{sign}(\hat{w}) = s)$ tends to a limit $\rho(s,\nu_0) \in (0,1)$, and we have:

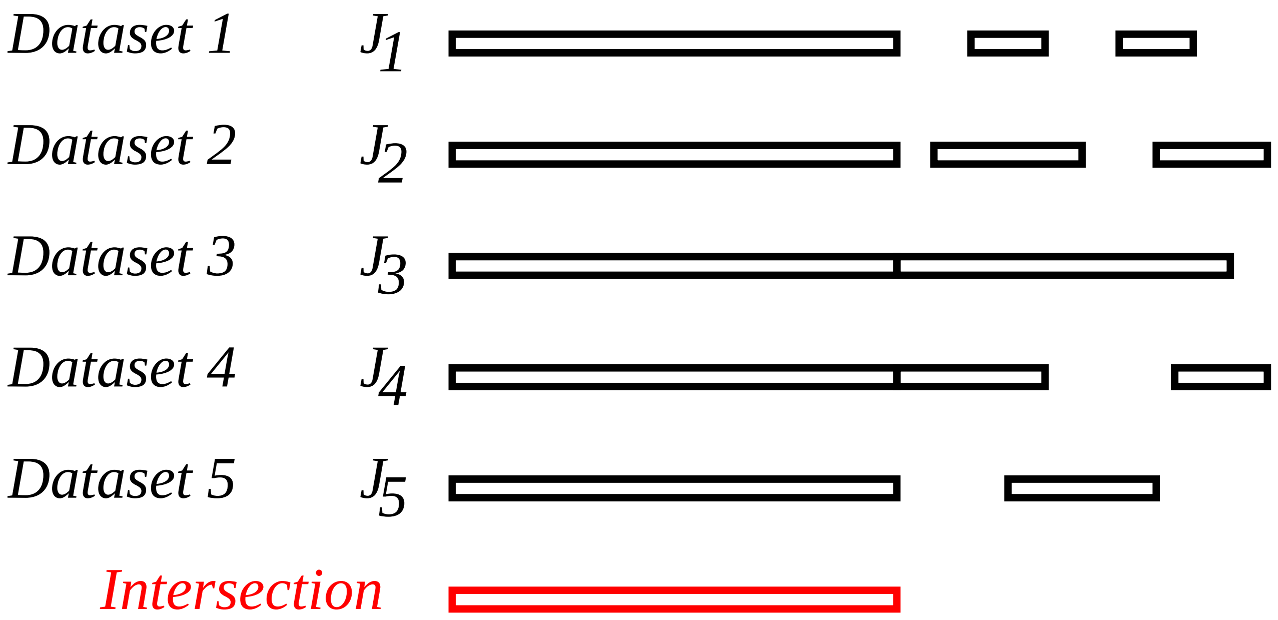
$$\mathbb{P}(\operatorname{sign}(\hat{w}) = s) - \rho(s,\nu_0) = O(n^{-1/2}\log n).$$

μ_n tends to zero at rate $n^{-1/2}$

- Summary of asymptotic behavior:
  - All relevant variables (i.e., the ones in $\mathbf{J}$) are selected with probability tending to one exponentially fast
  - All other variables are selected with strictly positive probability

μ_n tends to zero at rate $n^{-1/2}$

- Summary of asymptotic behavior:
  - All relevant variables (i.e., the ones in $\mathbf{J}$) are selected with probability tending to one exponentially fast
  - All other variables are selected with strictly positive probability
- If several datasets (with same distributions) are available, intersecting support sets would lead to the correct pattern with high probability



Bootstrap

- Given n i.i.d. observations $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$

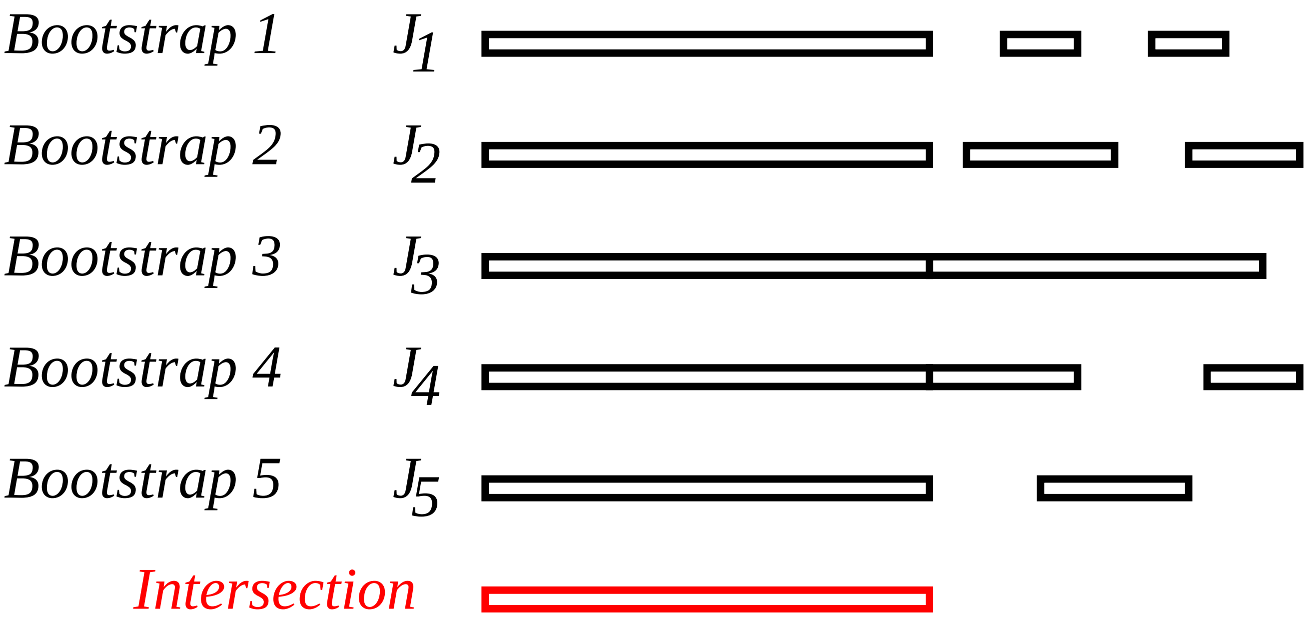
- m independent **bootstrap** replications: $k = 1, \ldots, m$,
  - *ghost samples* $(x_i^k, y_i^k) \in \mathbb{R}^p \times \mathbb{R}$, $i = 1, \ldots, n$, sampled independently and uniformly at random **with replacement** from the n original pairs

- Each bootstrap sample is composed of n potentially (and usually) duplicated copies of the original data pairs

- Standard way of mimicking availability of several datasets [ET98]

Bolasso algorithm

- m applications of the Lasso/Lars algorithm [EHJT04]
  - Intersecting supports of variables
  - Final estimation of w on the entire dataset



Bolasso - Consistency result

- **Proposition** [Bac08a]: Assume $\mu_n = \nu_0 n^{-1/2}$, with $\nu_0 > 0$. Then, for all $m > 1$, the probability that the Bolasso does not exactly select the correct model has the following upper bound:

$$\mathbb{P}(J \neq \mathbf{J}) \leqslant A_1 m e^{-A_2 n} + A_3 \frac{\log(n)}{n^{1/2}} + A_4 \frac{\log(m)}{m},$$

  where A_1, A_2, A_3, A_4 are strictly positive constants.

- Valid even if the Lasso consistency is not satisfied

- Influence of n, m

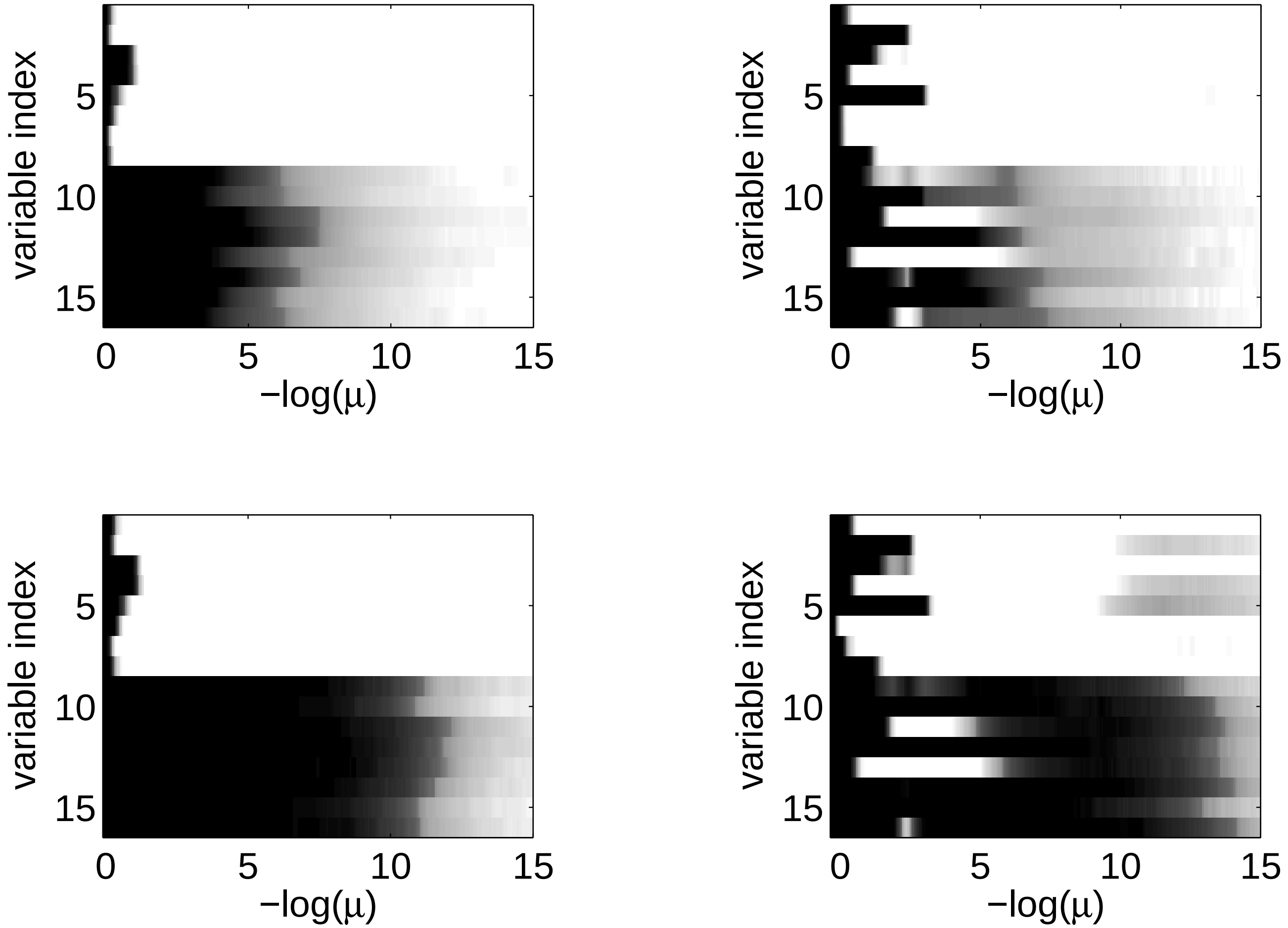
- Could be improved?

Consistency of the Lasso/Bolasso - Toy example

- Log-odd ratios of the probabilities of selection of each variable vs. μ

LASSO

BOLASSO



Consistency condition satisfied not satisfied

High-dimensional setting

- $p \geqslant n$: important case with harder analysis (no invertible covariance matrices)

- If consistency condition is satisfied, the Lasso is indeed consistent as long as $\log(p) << n$

- A lot of on-going work [MY08, Wai06]

High-dimensional setting (Lounici, 2008) [Lou08]

- Assumptions
  - $y_i = \mathbf{w}^\top x_i + \varepsilon_i$, ε i.i.d. normal with mean zero and variance σ^2
  - $Q = X^\top X/n$ with unit diagonal and cross-terms less than $\frac{1}{14s}$
  - **Theorem**: if $\|\mathbf{w}\|_0 \leqslant s$, and $A > 8^{1/2}$, then

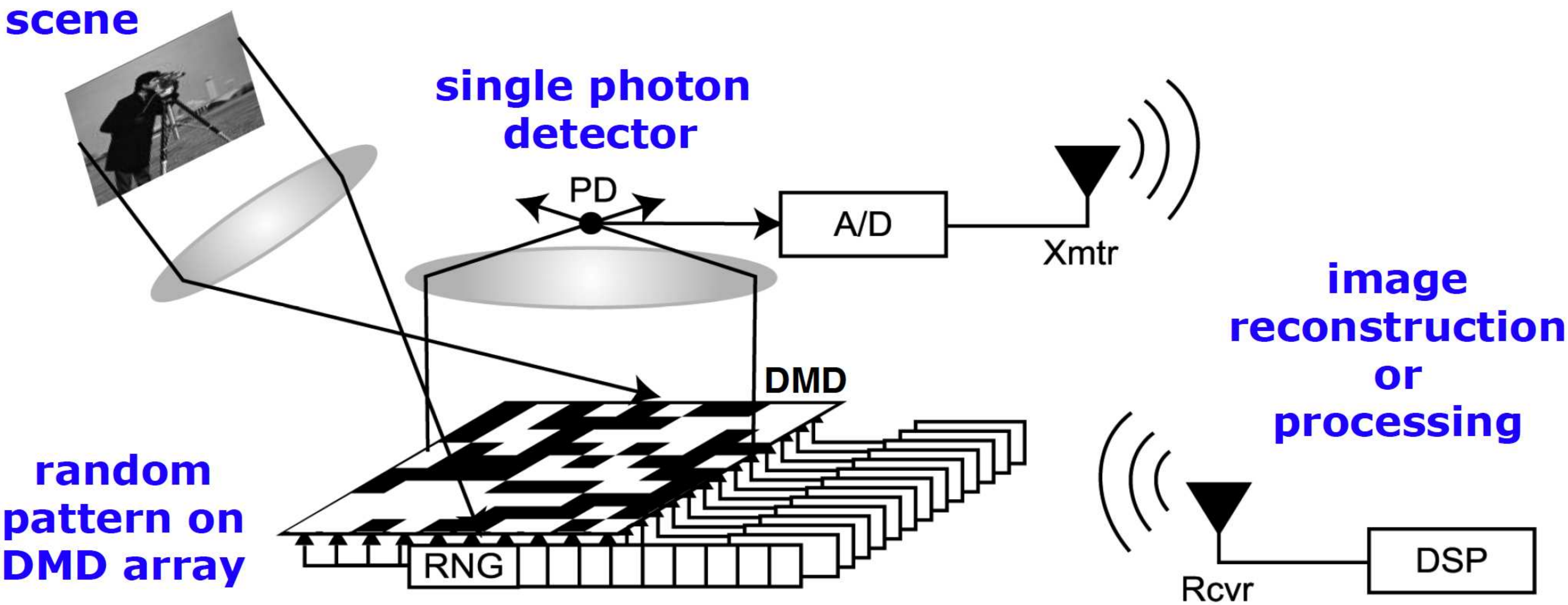
$$\mathbb{P}\left(\|\hat{w} - \mathbf{w}\|_\infty \leqslant 5A\sigma\left(\frac{\log p}{n}\right)^{1/2}\right) \leqslant 1 - p^{1-A^2/8}$$

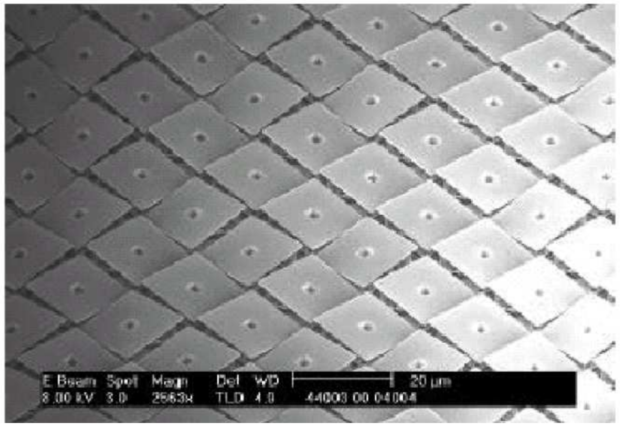
- Get the correct sparsity pattern if $\min_{j,\mathbf{w}_j\neq 0} |\mathbf{w}_j| > C\sigma\left(\frac{\log p}{n}\right)^{1/2}$
- Can have a lot of irrelevant variables!

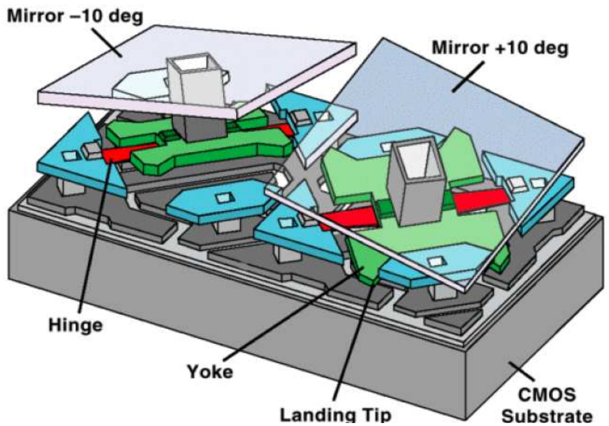
Links with compressed sensing [Bar07, CW08]

- Goal of compressed sensing: recover a signal $w \in \mathbb{R}^p$ from only n measurements $y = Xw \in \mathbb{R}^n$
- Assumptions: the signal is k-sparse, $n << p$
- Algorithm: $\min_{w \in \mathbb{R}^p} \|w\|_1$ such that $y = Xw$
- Sufficient condition on X and (k, n, p) for perfect recovery:
  - Restricted isometry property (all submatrices of $X^\top X$ must be well-conditioned)
  - that is, if $\|w\|_0 = k$, then $\|w\|_2(1-\delta_k) \leqslant \|Xw\|_2 \leqslant \|w\|_2(1+\delta_k)$
- Such matrices are hard to come up with deterministically, but random ones are OK with $k = \alpha p$, and $n/p = f(\alpha) < 1$

"Single-Pixel" CS Camera







w/ Kevin Kelly

Course Outline

1. ℓ^1**-norm regularization**
   - Review of nonsmooth optimization problems and algorithms
   - Algorithms for the Lasso (generic or dedicated)
   - Examples

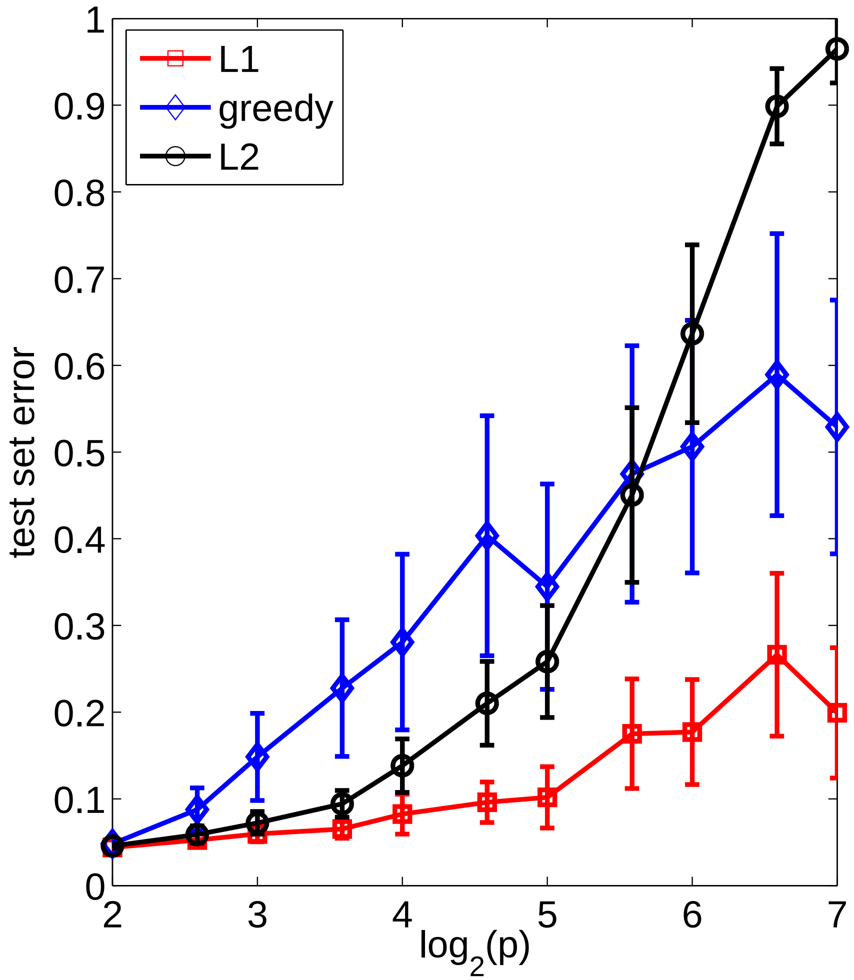
2. **Extensions**
   - Group Lasso and multiple kernel learning (MKL) + case study
   - Sparse methods for matrices
   - Sparse PCA

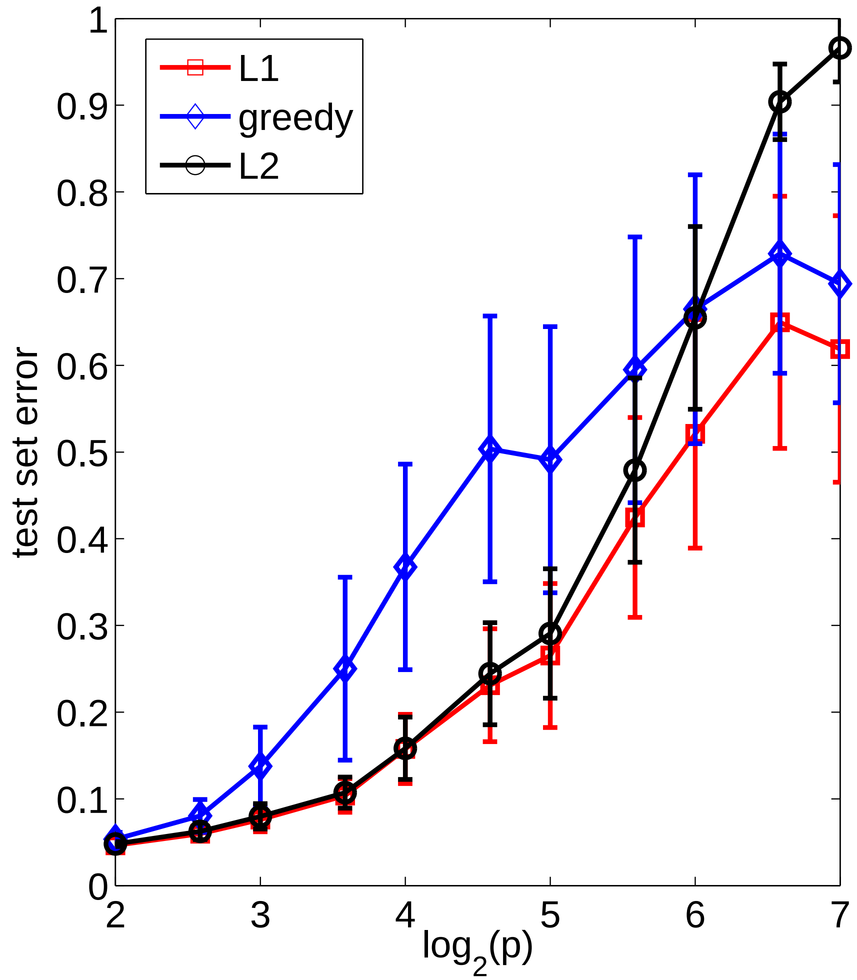
3. **Theory - Consistency of pattern selection**
   - Low and high dimensional setting
   - Links with compressed sensing

Summary - interesting problems

- Sparsity through non Euclidean norms
- Alternative approaches to sparsity
  - greedy approaches - Bayesian approaches
- Important (often non treated) question: when does sparsity actually help?
- Current research directions
  - Algorithms, algorithms, algorithms!
  - Design of good projections/measurement matrices for denoising or compressed sensing [See08]
  - Structured norm for structured situations (variables are usually not created equal) $\Rightarrow$ hierarchical Lasso or MKL[ZRY08, Bac08b]

Lasso in action





(left: sparsity is expected, right: sparsity is not expected)

Hierarchical multiple kernel learning (HKL) [Bac08b]

- Lasso or group Lasso, with exponentially many variables/kernels

- Main application:
  - **nonlinear variables selection** with $x \in \mathbb{R}^p$

$$k_{v_1,\dots,v_p}(x,y) = \prod_{j=1}^{p} \exp(-v_i \alpha (x_i - y_i)^2) = \prod_{j,\ v_j=1} \exp(-\alpha (x_i - y_i)^2)$$

    where $v \in \{0,1\}^p$
  - 2^p kernels! (as many as subsets of $\{1,\dots,p\}$)

- Learning sparse combination $\Leftrightarrow$ nonlinear variable selection

- Two questions:
  - Optimization in polynomial time?
  - Consistency?

Hierarchical multiple kernel learning (HKL) [Bac08b]

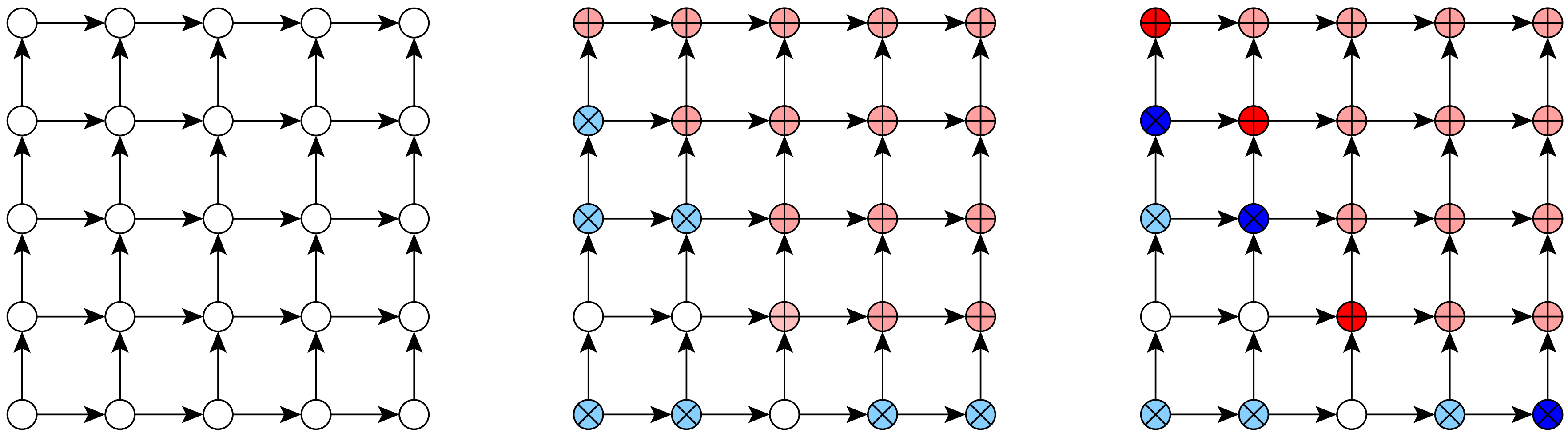
- The 2^p kernels are not created equal!

- Natural hierarchical structure (directed acyclic graph)
  - Goal: select a subset only after all of its subsets have been selected
  - Design a norm to achieve this behavior

$$\sum_{v \in V} \|\beta_{\text{descendants}(v)}\| = \sum_{v \in V} \left(\sum_{w \in \text{descendants}(v)} \|\beta_w\|^2 \right)^{1/2}$$

- Feature search algorithm in polynomial time in p and the number of selected kernels

Hierarchical multiple kernel learning (HKL) [Bac08b]



References

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, 68:337–404, 1950.

[Bac08a] F. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML)*, 2008.

[Bac08b] F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Adv. NIPS*, 2008.

[Bac08c] F. R. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, to appear, 2008.

[Bar07] Richard Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007.

[BB08] Léon Bottou and Olivier Bousquet. Learning using large datasets. In *Mining Massive DataSets for Security*, NATO ASI Workshop Series. IOS Press, Amsterdam, 2008. to appear.

[BGLS03] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizbal. *Numerical Optimization Theoretical and Practical Aspects*. Springer, 2003.

[BHH06] F. R. Bach, D. Heckerman, and E. Horvitz. Considering cost asymmetry in learning classifiers. *Journal of Machine Learning Research*, 7:1713–1741, 2006.

[BHK98] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, W.I., 1998. Morgan Kaufman.

[BL00] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization*. Number 3 in CMS Books in Mathematics. Springer-Verlag, 2000.

[BLJ04] 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)*, 2004.

[BTJ04] F. R. Bach, R. Thibaux, and M. I. Jordan. Computing regularization paths for learning multiple kernels. In *Advances in Neural Information Processing Systems 17*, 2004.

[BV03] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 2003.

[CDS01] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001.

[CW08] Emmanuel Candès and Michael Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[DGJL07] A. D'aspremont, El L. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–48, 2007.

[EA06] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Proc.*, 15(12):3736–3745, 2006.

[EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32:407, 2004.

[ET98] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1998.

[FHB01] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings American Control Conference*, volume 6, pages 4734–4739, 2001.

[GFW03] Thomas Gärtner, Peter A. Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *COLT*, 2003.

[HB07] 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.

[HCM+00] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *J. Mach. Learn. Res.*, 1:49–75, 2000.

[HRTZ05] 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.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[KW71] G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applicat.*, 33:82–95, 1971.

[LBC+04] 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, 2004.

[LBRN07] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *NIPS*, 2007.

[LCG+04] 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, 2004.

[Lou08] K. Lounici. Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators. *Electronic Journal of Statistics*, 2, 2008.

[MSE08] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *SIAM Multiscale Modeling and Simulation*, 7(1):214–241, 2008.

[MY08] N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Ann. Stat.*, page to appear, 2008.

[NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*, chapter 1. Springer, 2nd edition, 2006.

[OTJ07] G. Obozinski, B. Taskar, and M. I. Jordan. Multi-task feature selection. Technical report, UC Berkeley, 2007.

[PAE07] M. Pontil, A. Argyriou, and T. Evgeniou. Multi-task feature learning. In *Advances in Neural Information Processing Systems*, 2007.

[RBCG08] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. Simplemkl. *Journal of Machine Learning Research*, to appear, 2008.

[See08] M. Seeger. Bayesian inference and optimal design in the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.

[SMH07] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 791–798, New York, NY, USA, 2007. ACM.

[SRJ05] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, 2005.

[SRSS06] S. Sonnenbrug, G. Raetsch, C. Schaefer, and B. Schoelkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[SS01] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2001.

[STC04] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Camb. U. P., 2004.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of The Royal Statistical Society Series B*, 58(1):267–288, 1996.

[Wah90] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[Wai06] 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.

[WL08] Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.*, 2(1):224–244, 2008.

[YL07] M. Yuan and Y. Lin. On the non-negative garrotte estimator. *Journal of The Royal Statistical Society Series B*, 69(2):143–161, 2007.

[ZHT06] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comput. Graph. Statist.*, 15:265–286, 2006.

[Zou06] H. Zou. The adaptive lLsso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, December 2006.

[ZRY08] P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics*, To appear, 2008.

[ZY06] P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

Code

- ℓ^1-penalization: Matlab and R code available from `www.dsp.ece.rice.edu/cs`
- Multiple kernel learning:
  `asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html`
  `www.stat.berkeley.edu/~gobo/SKMsmo.tar`
- Other interesting code
  `www.shogun-toolbox.org`