Sparse methods for machine learning
Theory and algorithms

Francis Bach
Willow project, INRIA - Ecole Normale Supérieure

NIPS Tutorial - December 2009
Special thanks to R. Jenatton, J. Mairal, G. Obozinski
Supervised learning and regularization

• Data: \( x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, \ldots, n \)

• Minimize with respect to function \( f : \mathcal{X} \to \mathcal{Y} \):

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

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., $\ell_2$-norms)
     - Possibility of non linear predictors
     - Non parametric supervised learning and kernel methods
     - Well developed 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., $\ell_2$-norms)
     - Possibility of non linear predictors
     - Non parametric supervised learning and kernel methods
     - Well developed 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 $f(x) = w^T x$
     - Main example: $\ell_1$-norm $\|w\|_1 = \sum_{i=1}^{p} |w_i|$
     - Perform model selection as well as regularization
     - Theory and algorithms “in the making”
$\ell_2$ vs. $\ell_1$ - 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, 2009; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if

   \[ \| Q_{J^cJ} Q_{JJ}^{-1} \text{sign}(w_J) \|_\infty \leq 1, \]

   where \( Q = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top \in \mathbb{R}^{p \times p} \).
Lasso - Two main recent theoretical results

1. **Support recovery condition** (Zhao and Yu, 2006; Wainwright, 2009; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if

   \[ \| Q_{J^c} Q_{JJ}^{-1} \text{sign}(w_J) \|_\infty \leq 1, \]

   where \( Q = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top \in \mathbb{R}^{p \times p} \)

2. **Exponentially many irrelevant variables** (Zhao and Yu, 2006; Wainwright, 2009; 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

- $\ell_1$-norm for **linear** feature selection in **high dimensions**
  - Lasso usually not applicable directly

- **Non-linearities**

- **Dealing with exponentially many features**

- **Sparse learning on matrices**
Going beyond the Lasso

Non-linearity - Multiple kernel learning

• Multiple kernel learning
  – Learn sparse combination of matrices $k(x, x') = \sum_{j=1}^{p} \eta_j k_j(x, x')$
  – Mixing positive aspects of $\ell_1$-norms and $\ell_2$-norms

• Equivalent to group Lasso
  – $p$ multi-dimensional features $\Phi_j(x)$, where
    \[ k_j(x, x') = \Phi_j(x) \top \Phi_j(x') \]
  – learn predictor $\sum_{j=1}^{p} w_j \top \Phi_j(x)$
  – Penalization by $\sum_{j=1}^{p} \|w_j\|_2$
Going beyond the Lasso
Structured set of features

- Dealing with exponentially many features
  - Can we design efficient algorithms for the case $\log p \approx n$?
  - Use structure to reduce the number of allowed patterns of zeros
  - Recursivity, hierarchies and factorization

- Prior information on sparsity patterns
  - Grouped variables with overlapping groups
Going beyond the Lasso
Sparse methods on matrices

• **Learning problems on matrices**
  – Multi-task learning
  – Multi-category classification
  – Matrix completion
  – Image denoising
  – NMF, topic models, etc.

• **Matrix factorization**
  – Two types of sparsity (low-rank or dictionary learning)
Sparse methods for machine learning

Outline

• Introduction - Overview

• Sparse linear estimation with the $\ell_1$-norm
  – Convex optimization and algorithms
  – Theoretical results

• Structured sparse methods on vectors
  – Groups of features / Multiple kernel learning
  – Extensions (hierarchical or overlapping groups)

• Sparse methods on matrices
  – Multi-task learning
  – Matrix factorization (low-rank, sparse PCA, dictionary learning)
Why $\ell_1$-norm constraints leads to sparsity?

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

- Geometric interpretation
  - NB: penalizing is “equivalent” to constraining

![Diagram](image-url)
$\ell_1$-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 $w \in \mathbb{R}^p$:

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

  Error on data + 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)
A review of nonsmooth convex analysis and optimization

- **Analysis**: optimality conditions
- **Optimization**: algorithms
  - First-order methods
Optimality conditions for smooth optimization

Zero gradient

- Example: \( \ell_2 \)-regularization: 
  \[
  \min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \frac{\lambda}{2} \|w\|^2 \]

  - Gradient \( \nabla J(w) = \sum_{i=1}^{n} \ell'(y_i, w^\top x_i)x_i + \lambda w \) where \( \ell'(y_i, w^\top x_i) \) is the partial derivative of the loss w.r.t the second variable
  - If square loss, \( \sum_{i=1}^{n} \ell(y_i, w^\top x_i) = \frac{1}{2} \|y - Xw\|^2 \)
    * gradient = \(-X^\top(y - Xw) + \lambda w\)
    * normal equations \( \Rightarrow w = (X^\top X + \lambda I)^{-1} X^\top y \)
Optimality conditions for smooth optimization

Zero gradient

- Example: $\ell_2$-regularization: $\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \frac{\lambda}{2} \|w\|_2^2$

  - Gradient $\nabla J(w) = \sum_{i=1}^{n} \ell'(y_i, w^\top x_i)x_i + \lambda w$ where $\ell'(y_i, w^\top x_i)$ is the partial derivative of the loss w.r.t the second variable
  - If square loss, $\sum_{i=1}^{n} \ell(y_i, w^\top x_i) = \frac{1}{2} \|y - Xw\|_2^2$
    * gradient $= -X^\top(y - Xw) + \lambda w$
    * normal equations $\Rightarrow w = (X^\top X + \lambda I)^{-1}X^\top y$

- $\ell_1$-norm is non differentiable!
  - cannot compute the gradient of the absolute value
    $\Rightarrow$ Directional derivatives (or subgradient)
Directional derivatives - convex functions on $\mathbb{R}^p$

- Directional derivative in the direction $\Delta$ at $w$:
  \[
  \nabla J(w, \Delta) = \lim_{\varepsilon \to 0^+} \frac{J(w + \varepsilon \Delta) - J(w)}{\varepsilon}
  \]

- Always exist when $J$ is convex and continuous

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

- Proposition: $J$ is differentiable at $w$, if and only if $\Delta \mapsto \nabla J(w, \Delta)$ is linear. Then, $\nabla J(w, \Delta) = \nabla J(w)^\top \Delta$
Optimality conditions for convex functions

- Unconstrained minimization (function defined on $\mathbb{R}^p$):
  - **Proposition:** $w$ is optimal if and only if $\forall \Delta \in \mathbb{R}^p$, $\nabla J(w, \Delta) \geq 0$
  - Go up locally in all directions

- Reduces to zero-gradient for smooth problems

- Constrained minimization (function defined on a convex set $K$)
  - restrict $\Delta$ to directions so that $w + \varepsilon \Delta \in K$ for small $\varepsilon$
Directional derivatives for $\ell_1$-norm regularization

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

- $\ell_1$-norm: $\|w + \varepsilon \Delta\|_1 - \|w\|_1 = \sum_{j, w_j \neq 0} \{|w_j + \varepsilon \Delta_j| - |w_j|\} + \sum_{j, w_j = 0} |\varepsilon \Delta_j|$

- Thus,

$$\nabla J(w, \Delta) = \nabla L(w)^\top \Delta + \lambda \sum_{j, w_j \neq 0} \text{sign}(w_j) \Delta_j + \lambda \sum_{j, w_j = 0} |\Delta_j|$$

$$= \sum_{j, w_j \neq 0} [\nabla L(w)_j + \lambda \text{sign}(w_j)] \Delta_j + \sum_{j, w_j = 0} [\nabla L(w)_j \Delta_j + \lambda |\Delta_j|]$$

- Separability of optimality conditions
Optimality conditions for $\ell_1$-norm regularization

- **General loss:** $w$ optimal if and only if for all $j \in \{1, \ldots, p\}$,

  $\text{sign}(w_j) \neq 0 \quad \Rightarrow \quad \nabla L(w)_j + \lambda \text{sign}(w_j) = 0$

  $\text{sign}(w_j) = 0 \quad \Rightarrow \quad |\nabla L(w)_j| \leq \lambda$

- **Square loss:** $w$ optimal if and only if for all $j \in \{1, \ldots, p\}$,

  $\text{sign}(w_j) \neq 0 \quad \Rightarrow \quad -X_j^\top(y - Xw) + \lambda \text{sign}(w_j) = 0$

  $\text{sign}(w_j) = 0 \quad \Rightarrow \quad |X_j^\top(y - Xw)| \leq \lambda$

- For $J \subset \{1, \ldots, p\}$, $X_J \in \mathbb{R}^{n \times |J|} = X(:, J)$ denotes the columns of $X$ indexed by $J$, i.e., variables indexed by $J$
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) \( \alpha_t \)
  - 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: \( f(w_t) - f^* = O\left( \frac{M}{\sqrt{t}} \right) \)
  - and, differentiable with \( L \)-Lipschitz gradient: \( f(w_t) - f^* = O\left( \frac{L}{t} \right) \)
  - and, \( f \) \( \mu \)-strongly convex: \( f(w_t) - f^* = O\left( L \exp\left(-4t \frac{\mu}{L}\right)\right) \)

- \( \frac{\mu}{L} \) = condition number of the optimization problem

- **Coordinate descent:** similar properties

- **NB:** “optimal scheme” \( f(w_t) - f^* = O\left( L \min\{\exp(-4t \sqrt{\mu/L}), t^{-2}\} \right) \)
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 $\forall \Delta, g_t^\top \Delta \leq \nabla J(w_t, \Delta)$
    * 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): $f(w_t) - f^* = O\left(\frac{M}{\sqrt{t}}\right)$
Counter-example
Coordinate descent for nonsmooth objectives
Counter-example (Bertsekas, 1995)

Steepest descent for nonsmooth objectives

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

- Steepest descent starting from any \( x \) such that \( x_1 > |x_2| > (9/16)^2|x_1| \)
Sparsity-inducing norms
Using the structure of the problem

- Problems of the form
  \[ \min_{w \in \mathbb{R}^p} L(w) + \lambda \| w \| \] or
  \[ \min_{\| w \| \leq \mu} L(w) \]
- \( L \) smooth
- Orthogonal projections on the ball or the dual ball can be performed
  in semi-closed form, e.g., \( \ell_1 \)-norm (Maculan et al., 1999)
  or mixed \( \ell_1-\ell_2 \) (see, e.g., van den Berg et al., 2009)

- May use similar techniques than smooth optimization
  - Projected gradient descent
  - Proximal methods (Beck and Teboulle, 2009)
  - Dual ascent methods

- Similar convergence rates
  - depends on the condition number of the loss
Cheap (and not dirty) algorithms for all losses

- **Coordinate descent** (Fu, 1998; Wu and Lange, 2008; Friedman et al., 2007)
  - convergent **here** under reasonable assumptions! (Bertsekas, 1995)
  - separability of optimality conditions
  - equivalent to iterative thresholding
Cheap (and not dirty) algorithms for all losses

- **Coordinate descent** (Fu, 1998; Wu and Lange, 2008; Friedman et al., 2007)
  - convergent here under reasonable assumptions! (Bertsekas, 1995)
  - separability of optimality conditions
  - equivalent to iterative thresholding

- **“η-trick”** (Micchelli and Pontil, 2006; Rakotomamonjy et al., 2008; Jenatton et al., 2009b)
  
  - Notice that \( \sum_{j=1}^{p} |w_j| = \min_{\eta \geq 0} \frac{1}{2} \sum_{j=1}^{p} \left\{ \frac{w_j^2}{\eta_j} + \eta_j \right\} \)
  
  - Alternating minimization with respect to \( \eta \) (closed-form) and \( w \)
    (weighted squared \( \ell_2 \)-norm regularized problem)
Cheap (and not dirty) algorithms for all losses

- **Coordinate descent** (Fu, 1998; Wu and Lange, 2008; Friedman et al., 2007)
  - convergent **here** under reasonable assumptions! (Bertsekas, 1995)
  - separability of optimality conditions
  - equivalent to iterative thresholding

- “**η-trick**” (Micchelli and Pontil, 2006; Rakotomamonjy et al., 2008; Jenatton et al., 2009b)
  - Notice that \( \sum_{j=1}^{P} |w_j| = \min_{\eta \geq 0} \frac{1}{2} \sum_{j=1}^{P} \left\{ \frac{w_j^2}{\eta_j} + \eta_j \right\} \)
  - Alternating minimization with respect to \( \eta \) (closed-form) and \( w \)
    (weighted squared \( \ell_2 \)-norm regularized problem)

- **Dedicated algorithms that use sparsity** (active sets and homotopy methods)
Special case of square loss

- Quadratic programming formulation: minimize

\[
\frac{1}{2} \| y - Xw \|^2 + \lambda \sum_{j=1}^{p} (w_j^+ + w_j^-) \text{ such that } w = w^+ - w^-, \ w^+ \geq 0, \ w^- \geq 0
\]
Special case of square loss

- **Quadratic programming formulation**: minimize

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

- **generic toolboxes ⇒ 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 \( \frac{1}{2}\|y - X_Jw_J\|^2 + \lambda s_J^T w_J \) w.r.t. \( w_J \)
  where \( J = \{j, s_j \neq 0\} \).
- Closed form solution \( w_J = (X_J^T X_J)^{-1}(X_J^T y - \lambda s_J) \)

- **Algorithm**: “Guess” \( s \) and check optimality conditions
Optimality conditions for the sign vector $s$ (Lasso)

• For $s \in \{-1, 0, 1\}^p$ sign vector, $J = \{j, s_j \neq 0\}$ the nonzero pattern

• potential closed form solution: $w_J = (X_J^T X_J)^{-1}(X_J^T y - \lambda s_J)$ and $w_{Jc} = 0$

• $s$ is optimal if and only if
  – active variables: $\text{sign}(w_J) = s_J$
  – inactive variables: $\|X_{Jc}^T(y - X_J w_J)\|_\infty \leq \lambda$

• Active set algorithms (Lee et al., 2007; Roth and Fischer, 2008)
  – Construct $J$ iteratively by adding variables to the active set
  – Only requires to invert small linear systems
Homotopy methods for the square loss (Markowitz, 1956; Osborne et al., 2000; Efron et al., 2004)

• **Goal:** Get all solutions for all possible values of the regularization parameter $\lambda$

• Same idea as before: if the sign vector is known,

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

valid, as long as,

– sign condition: $\text{sign}(w_J^*(\lambda)) = s_J$

– subgradient condition: $\|X_J^\top c(X_Jw_J^*(\lambda) - y)\|_{\infty} \leq \lambda$

– this defines an interval on $\lambda$: the path is thus **piecewise affine**

• Simply need to find break points and directions
Piecewise linear paths

weights vs regularization parameter

0 0.1 0.2 0.3 0.4 0.5 0.6

-0.6 -0.4 -0.2 0 0.2 0.4 0.6
Algorithms for $\ell_1$-norms (square loss):
Gaussian hare vs. Laplacian tortoise

- Coordinate descent: $O(pn)$ per iterations for $\ell_1$ and $\ell_2$
- “Exact” algorithms: $O(kpn)$ for $\ell_1$ vs. $O(p^2n)$ for $\ell_2$
Additional methods - Softwares

- Many contributions in signal processing, optimization, machine learning
  - Proximal methods (Nesterov, 2007; Beck and Teboulle, 2009)
  - Extensions to stochastic setting (Bottou and Bousquet, 2008)

- Extensions to other sparsity-inducing norms

- **Softwares**
  - Many available codes
  - SPAMS (SPArse Modeling Software) - note difference with SpAM (Ravikumar et al., 2008)
    
    http://www.di.ens.fr/willow/SPAMS/
Sparse methods for machine learning

Outline

• Introduction - Overview

• Sparse linear estimation with the $\ell_1$-norm
  – Convex optimization and algorithms
  – Theoretical results

• Structured sparse methods on vectors
  – Groups of features / Multiple kernel learning
  – Extensions (hierarchical or overlapping groups)

• Sparse methods on matrices
  – Multi-task learning
  – Matrix factorization (low-rank, sparse PCA, dictionary learning)
Theoretical results - Square loss

- Main assumption: data generated from a certain sparse $w$

- Three main problems:

  1. **Regular consistency**: convergence of estimator $\hat{w}$ to $w$, i.e., $\|\hat{w} - w\|$ tends to zero when $n$ tends to $\infty$

  2. **Model selection consistency**: convergence of the sparsity pattern of $\hat{w}$ to the pattern $w$

  3. **Efficiency**: convergence of predictions with $\hat{w}$ to the predictions with $w$, i.e., $\frac{1}{n}\|X\hat{w} - Xw\|_2^2$ tends to zero

- Main results:

  - Condition for model consistency (support recovery)
  - High-dimensional inference
Model selection consistency (Lasso)

- Assume \( w \) sparse and denote \( J = \{ j, w_j \neq 0 \} \) the nonzero pattern.

- **Support recovery condition** (Zhao and Yu, 2006; Wainwright, 2009; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if
  \[
  \| Q_{J^cJ} Q_{JJ}^{-1} \text{sign}(w_J) \|_\infty \leq 1
  \]

where \( Q = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^n x_i x_i^\top \in \mathbb{R}^{p \times p} \) (covariance matrix).
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, 2009; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if
  \[
  \| \mathbf{Q}_{\mathbf{J}^c \mathbf{J}} \mathbf{Q}_{\mathbf{J} \mathbf{J}}^{-1} \text{sign}(\mathbf{w}_\mathbf{J}) \|_\infty \leq 1
  \]
  where $\mathbf{Q} = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top \in \mathbb{R}^{p \times p}$ (covariance matrix)

- Condition depends on $\mathbf{w}$ and $\mathbf{J}$ (may be relaxed)
  - may be relaxed by maximizing out $\text{sign}(\mathbf{w})$ 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 $w$ sparse and denote $J = \{ j, w_j \neq 0 \}$ the nonzero pattern

- **Support recovery condition** (Zhao and Yu, 2006; Wainwright, 2009; Zou, 2006; Yuan and Lin, 2007): the Lasso is sign-consistent if and only if

$$\| Q_{J^c J} Q_{J J}^{-1} \text{sign}(w_J) \|_\infty \leq 1$$

where $Q = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top \in \mathbb{R}^{p \times p}$ (covariance matrix)

- 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 $\ell_1$-norm: \[
  \min_{w \in \mathbb{R}^p} L(w) + \lambda \sum_{j=1}^{p} \frac{|w_j|}{|\hat{w}_j|^\alpha}
  \]
  
  – $\hat{w}$ estimator obtained from $\ell_2$ or $\ell_1$ 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 $\ell_0$ 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)
Bolasso (Bach, 2008a)

- **Property**: for a specific choice of regularization parameter $\lambda \approx \sqrt{n}$:
  - all variables in $J$ are always selected with high probability
  - all other ones selected with probability in $(0, 1)$

- Use the bootstrap to simulate several replications
  - Intersecting supports of variables
  - Final estimation of $w$ on the entire dataset

```
Bootstrap 1  J_1
Bootstrap 2  J_2
Bootstrap 3  J_3
Bootstrap 4  J_4
Bootstrap 5  J_5
```

*Intersection*
Model selection consistency of the Lasso/Bolasso

- probabilities of selection of each variable vs. regularization param. $\mu$

**LASSO**

**BOLASSO**

Support recovery condition: satisfied, not satisfied
Theoretical results usually assume that non-zero $w_j$ are large enough, i.e., $|w_j| \geq \sigma \sqrt{\frac{\log p}{n}}$

May include too many variables but still predict well

Oracle inequalities
  - Predict as well as the estimator obtained with the knowledge of $J$
  - Assume i.i.d. Gaussian noise with variance $\sigma^2$
  - We have:
    $$\frac{1}{n} \mathbb{E} \|X \hat{w}_{\text{oracle}} - Xw\|_2^2 = \frac{\sigma^2 |J|}{n}$$
High-dimensional inference
Variable selection without computational limits

• Approaches based on penalized criteria (close to BIC)

$$\min_{J \subset \{1, \ldots, p\}} \left\{ \min_{w_J \in \mathbb{R}^{|J|}} \| y - X_J w_J \|_2^2 \right\} + C \sigma^2 |J| (1 + \log \frac{p}{|J|})$$

• **Oracle inequality** if data generated by $\mathbf{w}$ with $k$ non-zeros (Massart, 2003; Bunea et al., 2007):

$$\frac{1}{n} \| X \hat{\mathbf{w}} - X \mathbf{w} \|_2^2 \leq C \frac{k \sigma^2}{n} (1 + \log \frac{p}{k})$$

• Gaussian noise - **No assumptions regarding correlations**

• **Scaling between dimensions**: $\frac{k \log p}{n}$ small

• Optimal in the minimax sense
High-dimensional inference
Variable selection with orthogonal design

- **Orthogonal design**: assume that $\frac{1}{n}X^\top X = I$

- **Lasso is equivalent to soft-thresholding** $\frac{1}{n}X^\top Y \in \mathbb{R}^p$
  
  - Solution: $\hat{w}_j = \text{soft-thresholding of } \frac{1}{n}X^\top_j y = w_j + \frac{1}{n}X^\top_j \varepsilon \text{ at } \frac{\lambda}{n}$

\[ \min_{w \in \mathbb{R}} \frac{1}{2}w^2 - wt + a|w| \]

Solution \( w = (|t| - a)^+ \text{ sign}(t) \)
High-dimensional inference
Variable selection with orthogonal design

- **Orthogonal design**: assume that \( \frac{1}{n}X^\top X = I \)

- Lasso is equivalent to soft-thresholding \( \frac{1}{n}X^\top Y \in \mathbb{R}^p \)
  - Solution: \( \hat{w}_j = \) soft-thresholding of \( \frac{1}{n}X_j^\top y = w_j + \frac{1}{n}X_j^\top \varepsilon \) at \( \frac{\lambda}{n} \)
  - Take \( \lambda = A\sigma\sqrt{n \log p} \)

- **Where does the** \( \log p = O(n) \) **come from?**
  - Expectation of the maximum of \( p \) Gaussian variables \( \approx \sqrt{\log p} \)
  - Union-bound:
    \[
    \mathbb{P}(\exists j \in J^c, |X_j^\top \varepsilon| \geq \lambda) \leq \sum_{j \in J^c} \mathbb{P}(|X_j^\top \varepsilon| \geq \lambda) \\
    \leq |J^c| e^{-\frac{\lambda^2}{2n\sigma^2}} \leq pe^{-\frac{A^2}{2} \log p} = p^{1-\frac{A^2}{2}}
    \]
High-dimensional inference (Lasso)

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

• Precise conditions on covariance matrix \( Q = \frac{1}{n} X^\top X \).
  – **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 \( Q \) has unit diagonal
Mutual incoherence (uniform low correlations)

- **Theorem** (Lounici, 2008):
  - \( y_i = \mathbf{w}^\top x_i + \varepsilon_i, \varepsilon \) i.i.d. normal with mean zero and variance \( \sigma^2 \)
  - \( Q = X^\top X/n \) with unit diagonal and cross-terms less than \( \frac{1}{14k} \)
  - if \( \|w\|_0 \leq k \), and \( A^2 > 8 \), then, with \( \lambda = A\sigma\sqrt{n \log p} \)
    \[
    \mathbb{P}\left( \|\hat{w} - w\|_\infty \leq 5A\sigma\left(\frac{\log p}{n}\right)^{1/2} \right) \geq 1 - p^{1-A^2/8}
    \]

- Model consistency by thresholding if \( \min_{j, w_j \neq 0} |w_j| > C\sigma\sqrt{\log p/n} \)

- Mutual incoherence condition depends *strongly* on \( k \)

- Improved result by averaging over sparsity patterns (Candès and Plan, 2009b)
Restricted eigenvalue conditions

- **Theorem** (Bickel et al., 2009):

  \[ \kappa(k)^2 = \min_{|J| \leq k} \Delta, \min_{\Delta, \|J_c\|_1 \leq \|\Delta J\|_1} \frac{\Delta^\top Q \Delta}{\|\Delta J\|_2^2} > 0 \]

  - assume \( \lambda = A\sigma\sqrt{n \log p} \) and \( A^2 > 8 \)
  - then, with probability \( 1 - p^{1 - A^2/8} \), we have

    \[
    \text{estimation error} \quad \|\hat{w} - w\|_1 \leq \frac{16A}{\kappa^2(k)} \sigma k \sqrt{\frac{\log p}{n}}
    \]

    \[
    \text{prediction error} \quad \frac{1}{n} \|X\hat{w} - Xw\|_2^2 \leq \frac{16A^2}{\kappa^2(k)} \frac{\sigma^2 k}{n} \log p
    \]

- Condition imposes a potentially hidden scaling between \((n, p, k)\)

- Condition always satisfied for \(Q = I\)
Checking sufficient conditions

- Most of the conditions are not computable in polynomial time

- Random matrices
  - Sample $X \in \mathbb{R}^{n \times p}$ from the Gaussian ensemble
  - Conditions satisfied with high probability for certain $(n, p, k)$
  - Example from Wainwright (2009): $n \geq Ck \log p$

- Checking with convex optimization
  - Relax conditions to convex optimization problems (d’Aspremont et al., 2008; Juditsky and Nemirovski, 2008; d’Aspremont and El Ghaoui, 2008)
  - Example: sparse eigenvalues $\min_{|J| \leq k} \lambda_{\text{min}}(QJJ)$
  - Open problem: verifiable assumptions still lead to weaker results
Sparse methods

Common extensions

- **Removing bias of the estimator**
  - Keep the active set, and perform *unregularized* restricted estimation (Candès and Tao, 2007)
  - Better theoretical bounds
  - Potential problems of robustness

- **Elastic net** (Zou and Hastie, 2005)
  - Replace $\lambda \|w\|_1$ by $\lambda \|w\|_1 + \varepsilon \|w\|_2^2$
  - Make the optimization strongly convex with unique solution
  - Better behavior with heavily correlated variables
Relevance of theoretical results

• Most results only for the square loss
  – Extend to other losses (Van De Geer, 2008; Bach, 2009b)

• Most results only for $\ell_1$-regularization
  – May be extended to other norms (see, e.g., Huang and Zhang, 2009; Bach, 2008b)

• Condition on correlations
  – very restrictive, far from results for BIC penalty

• Non sparse generating vector
  – little work on robustness to lack of sparsity

• Estimation of regularization parameter
  – No satisfactory solution $\Rightarrow$ open problem
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
Alternative sparse methods

Bayesian methods

• Lasso: minimize $\sum_{i=1}^{n} (y_i - w^\top x_i)^2 + \lambda\|w\|_1$

  – Equivalent to MAP estimation with Gaussian likelihood and factorized Laplace prior $p(w) \propto \prod_{j=1}^{p} e^{-\lambda|w_j|}$ (Seeger, 2008)

  – However, posterior puts zero weight on exact zeros

• Heavy-tailed distributions as a proxy to sparsity

  – Student distributions (Caron and Doucet, 2008)
  – Generalized hyperbolic priors (Archambeau and Bach, 2008)
  – Instance of automatic relevance determination (Neal, 1996)

• Mixtures of “Diracs” and another absolutely continuous distributions, e.g., “spike and slab” (Ishwaran and Rao, 2005)

• Less theory than frequentist methods
Comparing Lasso and other strategies for linear regression

- Compared methods to reach the least-square solution
  - Ridge regression: \( \min_{w \in \mathbb{R}^p} \frac{1}{2} \| y - Xw \|^2_2 + \frac{\lambda}{2} \| w \|^2_2 \)
  - Lasso: \( \min_{w \in \mathbb{R}^p} \frac{1}{2} \| y - Xw \|^2_2 + \lambda \| w \|_1 \)
  - 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
Summary

\( \ell_1 \)-norm regularization

- \( \ell_1 \)-norm regularization leads to nonsmooth optimization problems
  - analysis through directional derivatives or subgradients
  - optimization may or may not take advantage of sparsity

- \( \ell_1 \)-norm regularization allows high-dimensional inference

- Interesting problems for \( \ell_1 \)-regularization
  - Stable variable selection
  - Weaker sufficient conditions (for weaker results)
  - Estimation of regularization parameter (all bounds depend on the unknown noise variance \( \sigma^2 \))
Extensions

• **Sparse methods are not limited to the square loss**
  – e.g., theoretical results for logistic loss (Van De Geer, 2008; Bach, 2009b)

• **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
Questions?
Sparse methods for machine learning

Outline

• Introduction - Overview

• Sparse linear estimation with the $\ell_1$-norm
  – Convex optimization and algorithms
  – Theoretical results

• Structured sparse methods on vectors
  – Groups of features / Multiple kernel learning
  – Extensions (hierarchical or overlapping groups)

• Sparse methods on matrices
  – Multi-task learning
  – Matrix factorization (low-rank, sparse PCA, dictionary learning)
Penalization with grouped variables
(Yuan and Lin, 2006)

- Assume that \( \{1, \ldots, p\} \) is **partitioned** into \( m \) groups \( G_1, \ldots, G_m \)

- Penalization by \( \sum_{i=1}^{m} \| w_{G_i} \|_2 \), often called \( \ell_1-\ell_2 \) 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**
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**
  
  – $\ell_1$-norm: explicit features
  
  – $\ell_2$-norm: representer theorem allows to consider implicit features if their dot products can be computed easily (kernel methods)
Kernel methods: regularization by $\ell_2$-norm

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

- Optimization problem:
  $$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top \Phi(x_i)) + \frac{\lambda}{2} ||w||^2$$
Kernel methods: regularization by $\ell_2$-norm

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

- Optimization problem:

  $$\min_{w \in \mathbb{R}^p} \sum_{i=1}^n \ell(y_i, w^\top \Phi(x_i)) + \frac{\lambda}{2} \|w\|^2$$

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

  - 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 $x \in \mathcal{X}$, $\Phi_j(x) \in \mathcal{F}_j$ and $w_j \in \mathcal{F}_j$

• Replace the $\ell_1$-norm $\sum_{j=1}^{P} |w_j|$ by “block” $\ell_1$-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 (MKL)
(Lanckriet et al., 2004b; Bach et al., 2004a)

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

\[
\begin{align*}
\Phi_1(x) & \quad \mapsto \quad w_1 \\
\Phi_j(x) & \quad \mapsto \quad w_j \\
\Phi_p(x) & \quad \mapsto \quad w_p
\end{align*}
\]

- Generalized additive models (Hastie and Tibshirani, 1990)
Regularization for multiple features

\[ \Phi_1(x)^\top w_1 \]
\[ \vdots \quad \vdots \]
\[ x \rightarrow \Phi_j(x)^\top w_j \rightarrow w_1^\top \Phi_1(x) + \cdots + w_p^\top \Phi_p(x) \]
\[ \vdots \quad \vdots \]
\[ \Phi_p(x)^\top w_p \]

- 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

\[
\Phi_1(x)^\top w_1 \\
\vdots \quad \vdots \\
x \rightarrow \Phi_j(x)^\top w_j \rightarrow w_1^\top \Phi_1(x) + \cdots + w_p^\top \Phi_p(x) \\
\vdots \quad \vdots \\
\Phi_p(x)^\top w_p
\]

- 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 \( \ell_1 \)-norm:

1. Algorithms
2. Analysis of sparsity inducing properties (Ravikumar et al., 2008; Bach, 2008b)
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):

\[
G(K) = \min_{w \in \mathcal{F}} \sum_{i=1}^{n} \ell(y_i, w^\top \Phi(x_i)) + \frac{\lambda}{2} \|w\|_2^2
\]

\[
= \max_{\alpha \in \mathbb{R}^n} - \sum_{i=1}^{n} \ell_i^*(\lambda \alpha_i) - \frac{\lambda}{2} \alpha^\top K \alpha
\]

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 $\ell_1$-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 $\ell_1$-norm regularization is equivalent to minimizing with respect to $\eta$ the optimal value $G(\sum_{j=1}^{p} \eta_j K_j)$

- (sparse) weights $\eta$ obtained from optimality conditions

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

- Single optimization problem for learning both $\eta$ and $\alpha$
Proof of equivalence

\[
\min_{w_1, \ldots, w_p} \sum_{i=1}^{n} \ell(y_i, \sum_{j=1}^{p} w_j^\top \Phi_j(x_i)) + \lambda \left( \sum_{j=1}^{p} \|w_j\|_2 \right)^2
\]

\[
= \min_{w_1, \ldots, w_p} \min_{\sum j \eta_j = 1} \sum_{i=1}^{n} \ell(y_i, \sum_{j=1}^{p} w_j^\top \Phi_j(x_i)) + \lambda \sum_{j=1}^{p} \|w_j\|_2^2 \eta_j
\]

\[
= \min_{\sum j \eta_j = 1} \min_{\tilde{w}_1, \ldots, \tilde{w}_p} \sum_{i=1}^{n} \ell(y_i, \sum_{j=1}^{p} \eta_j^{1/2} \tilde{w}_j^\top \Phi_j(x_i)) + \lambda \sum_{j=1}^{p} \|\tilde{w}_j\|_2^2 \text{ with } \tilde{w}_j = w_j \eta_j^{-1/2}
\]

\[
= \min_{\sum j \eta_j = 1} \min_{\tilde{w}} \sum_{i=1}^{n} \ell(y_i, \tilde{w}^\top \Psi_\eta(x_i)) + \lambda \|\tilde{w}\|_2^2 \text{ with } \Psi_\eta(x) = (\eta_1^{1/2} \Phi_1(x), \ldots, \eta_p^{1/2} \Phi_p(x))
\]

- 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  \( \eta \geq 0 \))
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)
  – $\eta$-trick + cutting-planes (Sonnenburg et al., 2006)
  – $\eta$-trick + projected gradient descent (Rakotomamonjy et al., 2008)
  – Active set (Bach, 2008c)
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 $\Leftrightarrow \ell_2$-norm
  - Sparse combination $\Leftrightarrow \ell_1$-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 $\Leftrightarrow \ell_2$-norm
  – Sparse combination $\Leftrightarrow \ell_1$-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

• See NIPS 2009 workshop “Understanding MKL methods”
Sparse methods for machine learning

Outline

- Introduction - Overview

- Sparse linear estimation with the $\ell_1$-norm
  - Convex optimization and algorithms
  - Theoretical results

- Structured sparse methods on vectors
  - Groups of features / Multiple kernel learning
  - Extensions (hierarchical or overlapping groups)

- Sparse methods on matrices
  - Multi-task learning
  - Matrix factorization (low-rank, sparse PCA, dictionary learning)
Lasso - Two main recent theoretical results

1. **Support recovery condition**

2. **Exponentially many irrelevant variables:** under appropriate assumptions, consistency is possible as long as

   \[ \log p = O(n) \]
Lasso - Two main recent theoretical results

1. **Support recovery condition**

2. **Exponentially many irrelevant variables**: under appropriate assumptions, consistency is possible as long as

   \[
   \log p = O(n)
   \]

- Question: is it possible to build a sparse algorithm that can learn from more than $10^{80}$ features?
Lasso - Two main recent theoretical results

1. **Support recovery condition**

2. **Exponentially many irrelevant variables**: under appropriate assumptions, consistency is possible as long as

\[ \log p = O(n) \]

- Question: is it possible to build a sparse algorithm that can learn from more than \(10^{80}\) features?
  - Some type of recursivity/factorization is needed!
Hierarchical kernel learning (Bach, 2008c)

- Many kernels can be decomposed as a sum of many “small” kernels indexed by a certain set $V$: $k(x, x') = \sum_{v \in V} k_v(x, x')$

- Example with $x = (x_1, \ldots, x_q) \in \mathbb{R}^q$ ($\Rightarrow$ non-linear variable selection)
  
  - Gaussian/ANOVA kernels: $p = \#(V) = 2^q$

  $$
  \prod_{j=1}^{q} \left(1 + e^{-\alpha(x_j - x'_j)^2}\right) = \sum_{J \subset \{1, \ldots, q\}} \prod_{j \in J} e^{-\alpha(x_j - x'_j)^2} = \sum_{J \subset \{1, \ldots, q\}} e^{-\alpha \|x_J - x'_J\|_2^2}
  $$

  - NB: decomposition is related to Cosso (Lin and Zhang, 2006)

- Goal: learning sparse combination $\sum_{v \in V} \eta_v k_v(x, x')$

- Universally consistent non-linear variable selection requires all subsets
Restricting the set of active kernels

• With flat structure
  – Consider block $\ell_1$-norm: $\sum_{v \in V} d_v \| w_v \|_2$
  – cannot avoid being linear in $p = \#(V) = 2^q$

• Using the structure of the small kernels
  1. for computational reasons
  2. to allow more irrelevant variables
Restricting the set of active kernels

- $V$ is endowed with a directed acyclic graph (DAG) structure: select a kernel only after all of its ancestors have been selected.

- Gaussian kernels: $V = \text{power set of } \{1, \ldots, q\}$ with inclusion DAG
  - Select a subset only after all its subsets have been selected.
DAG-adapted norm (Zhao & Yu, 2008)

• Graph-based structured regularization

  – $D(v)$ is the set of descendants of $v \in V$:
    \[
    \sum_{v \in V} d_v \|w_{D(v)}\|_2^2 = \sum_{v \in V} d_v \left( \sum_{t \in D(v)} \|w_t\|_2^2 \right)^{1/2}
    \]

• Main property: If $v$ is selected, so are all its ancestors
DAG-adapted norm (Zhao & Yu, 2008)

• Graph-based structured regularization
  – $D(v)$ is the set of descendants of $v \in V$:
  \[
  \sum_{v \in V} d_v \| w_{D(v)} \|_2 = \sum_{v \in V} d_v \left( \sum_{t \in D(v)} \| w_t \|_2^2 \right)^{1/2}
  \]

• Main property: If $v$ is selected, so are all its ancestors

• Hierarchical kernel learning (Bach, 2008c):
  – polynomial-time algorithm for this norm
  – necessary/sufficient conditions for consistent kernel selection
  – Scaling between $p$, $q$, $n$ for consistency
  – Applications to variable selection or other kernels
Scaling between $p$, $n$ and other graph-related quantities

- $n$ = number of observations
- $p$ = number of vertices in the DAG
- $\text{deg}(V)$ = maximum out degree in the DAG
- $\text{num}(V)$ = number of connected components in the DAG

**Proposition** (Bach, 2009a): Assume consistency condition satisfied, Gaussian noise and data generated from a sparse function, then the support is recovered with high-probability as soon as:

$$\log \text{deg}(V) + \log \text{num}(V) = O(n)$$
Scaling between $p$, $n$
and other graph-related quantities

- $n$ = number of observations
- $p$ = number of vertices in the DAG
- $\text{deg}(V)$ = maximum out degree in the DAG
- $\text{num}(V)$ = number of connected components in the DAG

- **Proposition** (Bach, 2009a): Assume consistency condition satisfied, Gaussian noise and data generated from a sparse function, then the support is recovered with high-probability as soon as:

  \[ \log \text{deg}(V) + \log \text{num}(V) = O(n) \]

- **Unstructured case**: $\text{num}(V) = p \Rightarrow \log p = O(n)$

- **Power set of $q$ elements**: $\text{deg}(V) = q \Rightarrow \log q = \log \log p = O(n)$
Mean-square errors (regression)

<table>
<thead>
<tr>
<th>dataset</th>
<th>$n$</th>
<th>$p$</th>
<th>$k$</th>
<th>$#(V)$</th>
<th>L2</th>
<th>greedy</th>
<th>MKL</th>
<th>HKL</th>
</tr>
</thead>
<tbody>
<tr>
<td>abalone</td>
<td>4177</td>
<td>10</td>
<td>pol4</td>
<td>$\approx 10^7$</td>
<td>44.2±1.3</td>
<td>43.9±1.4</td>
<td>44.5±1.1</td>
<td>43.3±1.0</td>
</tr>
<tr>
<td>abalone</td>
<td>4177</td>
<td>10</td>
<td>rbf</td>
<td>$\approx 10^{10}$</td>
<td>43.0±0.9</td>
<td>45.0±1.7</td>
<td>43.7±1.0</td>
<td>43.0±1.1</td>
</tr>
<tr>
<td>boston</td>
<td>506</td>
<td>13</td>
<td>pol4</td>
<td>$\approx 10^9$</td>
<td>17.1±3.6</td>
<td>24.7±10.8</td>
<td>22.2±2.2</td>
<td>18.1±3.8</td>
</tr>
<tr>
<td>boston</td>
<td>506</td>
<td>13</td>
<td>rbf</td>
<td>$\approx 10^{12}$</td>
<td>16.4±4.0</td>
<td>32.4±8.2</td>
<td>20.7±2.1</td>
<td>17.1±4.7</td>
</tr>
<tr>
<td>pumadyn-32fh</td>
<td>8192</td>
<td>32</td>
<td>pol4</td>
<td>$\approx 10^{22}$</td>
<td>57.3±0.7</td>
<td>56.4±0.8</td>
<td>56.4±0.7</td>
<td>56.4±0.8</td>
</tr>
<tr>
<td>pumadyn-32fh</td>
<td>8192</td>
<td>32</td>
<td>rbf</td>
<td>$\approx 10^{31}$</td>
<td>57.7±0.6</td>
<td>72.2±22.5</td>
<td>56.5±0.8</td>
<td>55.7±0.7</td>
</tr>
<tr>
<td>pumadyn-32fm</td>
<td>8192</td>
<td>32</td>
<td>pol4</td>
<td>$\approx 10^{22}$</td>
<td>6.9±0.1</td>
<td>6.4±1.6</td>
<td>7.0±0.1</td>
<td>3.1±0.0</td>
</tr>
<tr>
<td>pumadyn-32fm</td>
<td>8192</td>
<td>32</td>
<td>rbf</td>
<td>$\approx 10^{31}$</td>
<td>5.0±0.1</td>
<td>46.2±51.6</td>
<td>7.1±0.1</td>
<td>3.4±0.0</td>
</tr>
<tr>
<td>pumadyn-32nh</td>
<td>8192</td>
<td>32</td>
<td>pol4</td>
<td>$\approx 10^{22}$</td>
<td>84.2±1.3</td>
<td>73.3±25.4</td>
<td>83.6±1.3</td>
<td>36.7±0.4</td>
</tr>
<tr>
<td>pumadyn-32nh</td>
<td>8192</td>
<td>32</td>
<td>rbf</td>
<td>$\approx 10^{31}$</td>
<td>56.5±1.1</td>
<td>81.3±25.0</td>
<td>83.7±1.3</td>
<td>35.5±0.5</td>
</tr>
<tr>
<td>pumadyn-32nm</td>
<td>8192</td>
<td>32</td>
<td>pol4</td>
<td>$\approx 10^{22}$</td>
<td>60.1±1.9</td>
<td>69.9±32.8</td>
<td>77.5±0.9</td>
<td>5.5±0.1</td>
</tr>
<tr>
<td>pumadyn-32nm</td>
<td>8192</td>
<td>32</td>
<td>rbf</td>
<td>$\approx 10^{31}$</td>
<td>15.7±0.4</td>
<td>67.3±42.4</td>
<td>77.6±0.9</td>
<td>7.2±0.1</td>
</tr>
</tbody>
</table>
Extensions to other kernels

- Extension to graph kernels, string kernels, pyramid match kernels

- Exploring large feature spaces with structured sparsity-inducing norms
  - Opposite view than traditional kernel methods
  - Interpretable models

- Other structures than hierarchies or DAGs
Grouped variables

• Supervised learning with known groups:
  – The $\ell_1$-$\ell_2$ norm
    \[
    \sum_{G \in G} \|w_G\|_2 = \sum_{G \in G} \left( \sum_{j \in G} w_j^2 \right)^{1/2}, \text{ with } G \text{ a partition of } \{1, \ldots, p\}
    \]
  – The $\ell_1$-$\ell_2$ norm sets to zero non-overlapping groups of variables (as opposed to single variables for the $\ell_1$ norm)
**Grouped variables**

- Supervised learning with known groups:
  - The $\ell_1$-$\ell_2$ norm

  \[
  \sum_{G \in G} \|w_G\|_2 = \sum_{G \in G} (\sum_{j \in G} w_j^2)^{1/2}, \text{ with } G \text{ a partition of } \{1, \ldots, p\}
  \]

  - The $\ell_1$-$\ell_2$ norm sets to zero non-overlapping groups of variables (as opposed to single variables for the $\ell_1$ norm).

- However, the $\ell_1$-$\ell_2$ norm encodes **fixed/static prior information**, requires to know in advance how to group the variables

- What happens if the set of groups $G$ is not a partition anymore?
Structured Sparsity (Jenatton et al., 2009a)

- When penalizing by the $\ell_1$-$\ell_2$ norm

\[
\sum_{G \in G} \|w_G\|_2 = \sum_{G \in G} \left( \sum_{j \in G} w_j^2 \right)^{1/2}
\]

- The $\ell_1$ norm induces sparsity at the group level:

  * Some $w_G$’s are set to zero
  - Inside the groups, the $\ell_2$ norm does not promote sparsity

- Intuitively, the zero pattern of $w$ is given by

\[
\{ j \in \{1, \ldots, p\}; \ w_j = 0 \} = \bigcup_{G \in G'} G \quad \text{for some } G' \subseteq G.
\]

- This intuition is actually true and can be formalized
Examples of set of groups $G$ (1/3)

- Selection of contiguous patterns on a sequence, $p = 6$

- $G$ is the set of blue groups

- Any union of blue groups set to zero leads to the selection of a contiguous pattern
Examples of set of groups $G$ (2/3)

- Selection of rectangles on a 2-D grids, $p = 25$

- $G$ is the set of blue/green groups (with their complements, not displayed)

- Any union of blue/green groups set to zero leads to the selection of a rectangle
Examples of set of groups $G$ (3/3)

- Selection of diamond-shaped patterns on a 2-D grids, $p = 25$
  
  ![Diamond patterns on 2-D grids](image)

- It is possible to extend such settings to 3-D space, or more complex topologies

- See applications later (sparse PCA)
Relationship between $G$ and Zero Patterns
(Jenatton, Audibert, and Bach, 2009a)

- $G \rightarrow$ Zero patterns:
  - by generating the union-closure of $G$

- Zero patterns $\rightarrow$ $G$:
  - Design groups $G$ from any union-closed set of zero patterns
  - Design groups $G$ from any intersection-closed set of non-zero patterns
Overview of other work on structured sparsity

- Specific hierarchical structure (Zhao et al., 2009; Bach, 2008c)

- **Union-closed** (as opposed to intersection-closed) family of nonzero patterns (Jacob et al., 2009; Baraniuk et al., 2008)

- Nonconvex penalties based on information-theoretic criteria with greedy optimization (Huang et al., 2009)
Sparse methods for machine learning

Outline

• **Introduction - Overview**

• **Sparse linear estimation with the $\ell_1$-norm**
  – Convex optimization and algorithms
  – Theoretical results

• **Structured sparse methods on vectors**
  – Groups of features / Multiple kernel learning
  – Extensions (hierarchical or overlapping groups)

• **Sparse methods on matrices**
  – Multi-task learning
  – Matrix factorization (low-rank, sparse PCA, dictionary learning)
Learning on matrices - Collaborative Filtering (CF)

• Given $n_\mathcal{X}$ “movies” $x \in \mathcal{X}$ and $n_\mathcal{Y}$ “customers” $y \in \mathcal{Y}$,

• predict the “rating” $z(x, y) \in \mathcal{Z}$ of customer $y$ for movie $x$

• Training data: large $n_\mathcal{X} \times n_\mathcal{Y}$ incomplete matrix $\mathcal{Z}$ that describes the known ratings of some customers for some movies

• Goal: complete the matrix.
Learning on matrices - Multi-task learning

- $k$ prediction tasks on same covariates $x \in \mathbb{R}^p$
  - $k$ weight vectors $w_j \in \mathbb{R}^p$
  - Joint matrix of predictors $W = (w_1, \ldots, w_k) \in \mathbb{R}^{p \times k}$

- Many applications
  - “transfer learning”
  - Multi-category classification (one task per class) (Amit et al., 2007)

- Share parameters between various tasks
  - similar to fixed effect/random effect models (Raudenbush and Bryk, 2002)
  - joint variable or feature selection (Obozinski et al., 2009; Pontil et al., 2007)
Learning on matrices - Image denoising

- Simultaneously denoise all patches of a given image
- Example from Mairal, Bach, Ponce, Sapiro, and Zisserman (2009c)
Two types of sparsity for matrices $M \in \mathbb{R}^{n \times p}$

I - Directly on the elements of $M$

- Many zero elements: $M_{ij} = 0$

- Many zero rows (or columns): $(M_{i1}, \ldots, M_{ip}) = 0$
Two types of sparsity for matrices $M \in \mathbb{R}^{n \times p}$

II - Through a factorization of $M = UV^T$

- $M = UV^T$, $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{n \times m}$

- **Low rank:** $m$ small

- **Sparse decomposition:** $U$ sparse
Structured matrix factorizations - Many instances

- $M = UV^\top$, $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{p \times m}$

- **Structure on $U$ and/or $V$**
  - Low-rank: $U$ and $V$ have few columns
  - Dictionary learning / sparse PCA: $U$ or $V$ has many zeros
  - Clustering ($k$-means): $U \in \{0, 1\}^{n \times m}$, $U1 = 1$
  - Pointwise positivity: non negative matrix factorization (NMF)
  - Specific patterns of zeros
  - etc.

- **Many applications**
  - e.g., source separation (Févotte et al., 2009), exploratory data analysis
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 $M \in \mathbb{R}^{n \times p}$
  
  – Rank of $M$ is the minimum size $m$ of all factorizations of $M$ into $M = UV^\top$, $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{p \times m}$
  – Singular value decomposition: $M = U \text{Diag}(s)V^\top$ where $U$ and $V$ have orthonormal columns and $s \in \mathbb{R}_+^m$ are singular values

• Rank of $M$ equal to the number of non-zero singular values
Low-rank matrix factorizations

**Trace norm**

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

- Rank recovery condition (Bach, 2008d)
  - The Hessian of the loss around the asymptotic solution should be close to diagonal

- Sufficient condition for exact rank minimization (Recht et al., 2009)

- High-dimensional inference for noisy matrix completion (Srebro et al., 2005; Candès and Plan, 2009a)
  - May recover entire matrix from slightly more entries than the minimum of the two dimensions

- **Efficient algorithms**:
  - First-order methods based on the singular value decomposition (see, e.g., Mazumder et al., 2009)
  - Low-rank formulations (Rennie and Srebro, 2005; Abernethy et al., 2009)
Spectral regularizations

• Extensions to any functions of singular values

• Extensions to bilinear forms (Abernethy et al., 2009)

\[(x, y) \mapsto \Phi(x)^\top B \Psi(y)\]

on features \(\Phi(x) \in \mathbb{R}^{f_X}\) and \(\Psi(y) \in \mathbb{R}^{f_Y}\), and \(B \in \mathbb{R}^{f_X \times f_Y}\)

• Collaborative filtering with attributes

• **Representer theorem**: the solution must be of the form

\[B = \sum_{i=1}^{n_X} \sum_{j=1}^{n_Y} \alpha_{ij} \Psi(x_i) \Phi(y_j)^\top\]

• Only norms invariant by orthogonal transforms (Argyriou et al., 2009)
Sparse principal component analysis

- Given data matrix \( X = (x_1^\top, \ldots, x_n^\top)^\top \in \mathbb{R}^{n \times p} \), principal component analysis (PCA) may be seen from two perspectives:
  - **Analysis view**: find the projection \( v \in \mathbb{R}^p \) of maximum variance (with deflation to obtain more components)
  - **Synthesis view**: find the basis \( v_1, \ldots, v_k \) such that all \( x_i \) 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 different
**Sparse principal component analysis**

**Analysis view**

- **DSPCA** (d’Aspremont et al., 2007), with $A = \frac{1}{n}X^\top X \in \mathbb{R}^{p \times p}$

\[
\max_{\|v\|_2 = 1, \|v\|_0 \leq k} v^\top Av \quad \text{relaxed into} \quad \max_{\|v\|_2 = 1, \|v\|_1 \leq k^{1/2}} v^\top Av
\]

- using $M = vv^\top$, itself relaxed into

\[
\max_{M \succeq 0, \text{tr } M = 1, 1^\top |M| 1 \leq k} \text{tr } AM
\]
### Sparse principal component analysis

**Analysis view**

- **DSPCA** (d’Aspremont et al., 2007), with $A = \frac{1}{n}X^\top X \in \mathbb{R}^{p \times p}$

\[
\begin{align*}
\max_{\|v\|_2=1,\|v\|_0\leq k} v^\top Av & \quad \text{relaxed into} \quad \max_{\|v\|_2=1,\|v\|_1\leq k^{1/2}} v^\top Av \\
\text{using } M = vv^\top, \text{ itself relaxed into} & \quad \max_{M \succeq 0, \text{tr } M=1, 1^\top |M| \leq k} \text{tr } AM
\end{align*}
\]

- Requires deflation for multiple components (Mackey, 2009)
- More refined convex relaxation (d’Aspremont et al., 2008)
- Non convex analysis (Moghaddam et al., 2006b)
- Applications beyond interpretable principal components
  - used as sufficient conditions for high-dimensional inference
Sparse principal component analysis
Synthesis view

• Find $v_1, \ldots, v_m \in \mathbb{R}^p$ sparse so that

$$\sum_{i=1}^{n} \min_{u \in \mathbb{R}^m} \left\| x_i - \sum_{j=1}^{m} u_j v_j \right\|_2^2$$

is small

• Equivalent to look for $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{p \times m}$ such that $V$ is sparse and $\|X - UV^\top\|_F^2$ is small
Sparse principal component analysis

Synthesis view

• Find $v_1, \ldots, v_m \in \mathbb{R}^p$ sparse so that

\[
\sum_{i=1}^{n} \min_{u \in \mathbb{R}^m} \left\| x_i - \sum_{j=1}^{m} u_j v_j \right\|_2^2 \text{ is small}
\]

• Equivalent to look for $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{p \times m}$ such that $V$ is sparse and $\| X - UV^\top \|_F^2$ is small

• Sparse formulation (Witten et al., 2009; Bach et al., 2008)
  – Penalize columns $v_i$ of $V$ by the $\ell_1$-norm for sparsity
  – Penalize columns $u_i$ of $U$ by the $\ell_2$-norm to avoid trivial solutions

\[
\min_{U,V} \| X - UV^\top \|_F^2 + \lambda \sum_{i=1}^{m} \left\{ \| u_i \|_2^2 + \| v_i \|_1^2 \right\}
\]
Structured matrix factorizations

\[
\min_{U,V} \|X - UV^\top\|^2_F + \lambda \sum_{i=1}^m \{\|u_i\|^2 + \|v_i\|^2\}
\]

- Penalizing by \(\|u_i\|^2 + \|v_i\|^2\) equivalent to constraining \(\|u_i\| \leq 1\) and penalizing by \(\|v_i\|\) (Bach et al., 2008)

- Optimization by alternating minimization (non-convex)

- \(u_i\) decomposition coefficients (or “code”), \(v_i\) dictionary elements

- **Sparse PCA** = sparse dictionary (\(\ell_1\)-norm on \(u_i\))

- **Dictionary learning** = sparse decompositions (\(\ell_1\)-norm on \(v_i\))
  - Olshausen and Field (1997); Elad and Aharon (2006); Raina et al. (2007)
Dictionary learning for image denoising

\[ x = x_0 + \varepsilon \]

measurements = original image + noise
Dictionary learning for image denoising

- **Solving the denoising problem** (Elad and Aharon, 2006)
  - Extract all overlapping $8 \times 8$ patches $x_i \in \mathbb{R}^{64}$.
  - Form the matrix $X = [x_1, \ldots, x_n] \top \in \mathbb{R}^{n \times 64}$
  - Solve a matrix factorization problem:
    \[
    \min_{U,V} \|X - UV\top\|_F^2 = \sum_{i=1}^{n} \|x_i - VU(i,:\|_2^2
    \]
    where $U$ is **sparse**, and $V$ is the **dictionary**
  - Each patch is decomposed into $x_i = VU(i,:)$
  - Average the reconstruction $VU(i,:)$ of each patch $x_i$ to reconstruct a full-sized image

- The number of patches $n$ is large (= number of pixels)
Online optimization for dictionary learning

\[
\min_{U \in \mathbb{R}^{n \times m}, V \in \mathcal{C}} \sum_{i=1}^{n} \|x_i - VU(i,:)\|_2^2 + \lambda \|U(i,:)\|_1
\]

\[
\mathcal{C} \overset{\Delta}{=} \{ V \in \mathbb{R}^{p \times m} \text{ s.t. } \forall j = 1, \ldots, m, \|V(:,j)\|_2 \leq 1 \}.
\]

- Classical optimization alternates between \( U \) and \( V \)
- Good results, but very slow!
Online optimization for dictionary learning

\[
\min_{U \in \mathbb{R}^{n \times m}, V \in \mathcal{C}} \sum_{i=1}^{n} \| x_i - VU(i,:) \|_2^2 + \lambda \| U(i,:) \|_1
\]

\[\mathcal{C} \triangleq \{ V \in \mathbb{R}^{p \times m} \text{ s.t. } \forall j = 1, \ldots, m, \| V(:,j) \|_2 \leq 1 \}.\]

- Classical optimization alternates between $U$ and $V$.
- Good results, but very slow!
- **Online learning** (Mairal, Bach, Ponce, and Sapiro, 2009a) can
  - handle potentially infinite datasets
  - adapt to dynamic training sets
Denoising result
(Mairal, Bach, Ponce, Sapiro, and Zisserman, 2009c)
Denoising result

(Mairal, Bach, Ponce, Sapiro, and Zisserman, 2009c)
What does the dictionary $V$ look like?
Inpainting a 12-Mpixel photograph

THE SALINAS VALLEY is in Northern California. It is a long narrow valley between two ranges of mountains, and the Salinas River winds and twists up the center until it dips at last into Monterey Bay.

I remember my childhood name for grasses and seasonal flowers. I remember where a toad may live and what time the birds migrated in the summer and what trees and seasons smelled like: how people looked and walked and worked and slept even. The memory of odor is very rich.

I remember that the Cobden Mountains to the east of the valley were high, snow-capped mountains filled of ice and sharpness and a kind of mystery, so that one wanted to climb into their warm interiors almost as you want to climb into the lap of a beloved mother. They were mountainous mountains with a low, grass-covered slope. The Santa Lucias stood up against the sky to the west and kept the valley from the open sea, and they were dark and brooding, unfriendly and dangerous. I always found in myself a sense of west and a love of east. Where I was, got such an idea I cannot say, unless it could be that the morning came over the peaks of the Cobdenes and the night drifted back from the slopes of the Santa Lucias. It may be that the birth and death of the day had some part in my feeling about the two ranges of mountains.

From both sides of the valley little streams slipped out of the hill canyons and fell into the bed of the Salinas River. In the winter of wet years the streams ran full-freash, and they swelled the river until sometimes it raged and boated, bank full, and then it was a destroyer. The river tore the edges of the farm lands and washed whole acres clean; it topped barns and houses into itself, to go floating and bobbing away. It trapped cows and pigs and sheep and drowned them in its muddy green water and carried them to the sea. Then when the late spring came, the river drew down from its edges and the land parks appeared. And in the summer the river didn't run its gantmore ground. Some pools would be left in the deep swale places under a high bank. The lilies and grasses grew back, and willows straightened up with the heel-deads in their upper branches. The Salinas was only a quiet time river. The waterer used to drive it underground. It was not a real river at all, but it was the only one we had and so we boasted about it how dangerous it was in a wet winter and how dry it was in a dry summer. You just boast about anything if it's all you have. Maybe the less you have, the more you are required to boast.

The floor of the Salinas Valley, between the ranges and below the foothills, is level because this valley used to be the bottom of a hundred-mile inlet from the sea. The river mouth at Moss Landing was centuries ago the entrance to this long inland water. Once, fifty miles down the valley, my father bored a well. The driller came up first with sand, then with gravel and then with white sea sand full of shells and even pl...
Inpainting a 12-Mpixel photograph
Inpainting a 12-Mpixel photograph
Inpainting a 12-Mpixel photograph
Alternative usages of dictionary learning

• Uses the “code” $U$ as representation of observations for subsequent processing (Raina et al., 2007; Yang et al., 2009)

• Adapt dictionary elements to specific tasks (Mairal et al., 2009b)
  – Discriminative training for weakly supervised pixel classification (Mairal et al., 2008)
Sparse Structured PCA  
(Jenatton, Obozinski, and Bach, 2009b)

• Learning **sparse and structured** dictionary elements:

\[
\min_{U,V} \|X - UV^\top\|_F^2 + \lambda \sum_{i=1}^{m} \left\{ \|u_i\|^2 + \|v_i\|^2 \right\}
\]

• Structured norm on the dictionary elements
  
  – grouped penalty with overlapping groups to select specific classes of sparsity patterns
  
  – use prior information for better reconstruction and/or added robustness

• Efficient learning procedures through $\eta$-tricks (closed form updates)
Application to face databases (1/3)

• NMF obtains partially local features
Application to face databases (2/3)

- Enforce selection of **convex** nonzero patterns \( \Rightarrow \) robustness to occlusion

(unstructured) sparse PCA  Structured sparse PCA
Application to face databases (2/3)

- Enforce selection of convex nonzero patterns ⇒ robustness to occlusion
Application to face databases (3/3)

- Quantitative performance evaluation on classification task
Latent Dirichlet allocation (Blei et al., 2003)

- For a document, sample $\theta \in \mathbb{R}^k$ from a Dirichlet($\alpha$)
- For the $n$-th word of the same document,
  * sample a topic $z_n$ from a multinomial with parameter $\theta$
  * sample a word $w_n$ from a multinomial with parameter $\beta(z_n, :)$
• **Latent Dirichlet allocation** (Blei et al., 2003)
  
  – For a document, sample \( \theta \in \mathbb{R}^k \) from a Dirichlet(\( \alpha \))
  
  – For the \( n \)-th word of the same document,
    * sample a topic \( z_n \) from a multinomial with parameter \( \theta \)
    * sample a word \( w_n \) from a multinomial with parameter \( \beta(z_n, :) \)

• **Interpretation as multinomial PCA** (Buntine and Perttu, 2003)

  – Marginalizing over topic \( z_n \), given \( \theta \), each word \( w_n \) is selected from a multinomial with parameter \( \sum_{z=1}^{k} \theta_k \beta(z, :) = \beta^\top \theta \)
  
  – Row of \( \beta = \) dictionary elements, \( \theta \) code for a document
Topic models and matrix factorization

- **Two different views on the same problem**
  - Interesting parallels to be made
  - Common problems to be solved

- **Structure on dictionary/decomposition coefficients** with adapted priors, e.g., nested Chinese restaurant processes (Blei et al., 2004)

- Other priors and probabilistic formulations (Griffiths and Ghahramani, 2006; Salakhutdinov and Mnih, 2008; Archambeau and Bach, 2008)

- **Identifiability and interpretation/evaluation of results**

- **Discriminative tasks** (Blei and McAuliffe, 2008; Lacoste-Julien et al., 2008; Mairal et al., 2009b)

- **Optimization and local minima**
Sparsifying linear methods

- Same pattern than with kernel methods
  - High-dimensional inference rather than non-linearities

- Main difference: in general no unique way

- Sparse CCA (Sriperumbudur et al., 2009; Hardoon and Shawe-Taylor, 2008; Archambeau and Bach, 2008)

- Sparse LDA (Moghaddam et al., 2006a)

- Sparse ...
Sparse methods for matrices

Summary

• Structured matrix factorization has many applications

• Algorithmic issues
  – Dealing with large datasets
  – Dealing with structured sparsity

• Theoretical issues
  – Identifiability of structures, dictionaries or codes
  – Other approaches to sparsity and structure

• Non-convex optimization versus convex optimization
  – Convexification through unbounded dictionary size (Bach et al., 2008; Bradley and Bagnell, 2009) - few performance improvements
Sparse methods for machine learning

Outline

• Introduction - Overview

• Sparse linear estimation with the $\ell_1$-norm
  – Convex optimization and algorithms
  – Theoretical results

• Structured sparse methods on vectors
  – Groups of features / Multiple kernel learning
  – Extensions (hierarchical or overlapping groups)

• Sparse methods on matrices
  – Multi-task learning
  – Matrix factorization (low-rank, sparse PCA, dictionary learning)
Links with compressed sensing
(Baraniuk, 2007; Candès and Wakin, 2008)

• 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, \( n \) much smaller than \( p \)

• 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 small submatrices of \( X^\top X \) must be well-conditioned)
  – Such matrices are hard to come up with deterministically, but random ones are OK with \( k \log p = O(n) \)

• Random \( X \) for machine learning?
Why use sparse methods?

• Sparsity as a proxy to interpretability
  – Structured sparsity

• 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
  – Problems are sparse if you make them sparse
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?
Hiring postdocs and PhD students

European Research Council project on
Sparse structured methods for machine learning

• PhD positions

• 1-year and 2-year postdoctoral positions

• Machine learning (theory and algorithms), computer vision, audio processing, signal processing

• Located in downtown Paris (Ecole Normale Supérieure - INRIA)

• http://www.di.ens.fr/~fbach/sierra/
References


M. Fazel, H. Hindi, and S.P. Boyd. A rank minimization heuristic with application to minimum


J. Liu, S. Ji, and J. Ye. Multi-Task Feature Learning Via Efficient $l_2$-Norm Minimization. *Proceedings*
of the 25th Conference on Uncertainty in Artificial Intelligence (UAI), 2009.


H. M. Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval


