On the Geometry of Optimization Problems and their Structure

PhD defense - Vincent Roulet

Supervised by Alexandre d'Aspremont

December 21, 2017

Part 1

Sharpness in Convex Optimization Problems

Optimization Problems

Goal Make best possible action for a given task

Optimization Problems

Goal Make best possible action for a given task

Classical Example: Minimize production cost of an item

Optimization Problems

Goal Make best possible action for a given task

Classical Example: Minimize production cost of an item

Formally,

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{C} \end{aligned}$$

in x where

- $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ represents the parameters of the task
- $f : \mathbb{R}^d \to \mathbb{R}$ measures the cost of an action
- $\mathcal{C} \subset \mathbb{R}^d$ represents constraints on the possible parameters

Optimization Problems

Goal Make best possible action for a given task

Classical Example: Minimize production cost of an item

Formally,

$$\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{C} \end{array}$$

in x where

- $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ represents the parameters of the task
- $f : \mathbb{R}^d \to \mathbb{R}$ measures the cost of an action
- $\mathcal{C} \subset \mathbb{R}^d$ represents constraints on the possible parameters

Applications

- Design an electronic circuit
- Fit a model to data (Machine Learning in 2nd part)
- ...

Optimization Algorithms

Optimization Algorithms

Principle: Search iteratively an approximate solution

Optimization Algorithms

Principle: Search iteratively an approximate solution

Description

1. Starts from $x_0 \in \mathcal{C}$
2. At each $t \geq 0$, gets information I_t on the problem at x_t
3. Builds x_{t+1} from previous information $\{I_0, \ldots, I_t\}$

- Assume here first order information $I_t = \{f(x_t), \nabla f(x_t)\}$
- Rule to design

Optimization Algorithms

Principle: Search iteratively an approximate solution

Description

1. Starts from $x_0 \in \mathcal{C}$
2. At each $t \geq 0$, gets information I_t on the problem at x_t
3. Builds x_{t+1} from previous information $\{I_0, \ldots, I_t\}$

- Assume here first order information $I_t = \{f(x_t), \nabla f(x_t)\}$
- Rule to design

Performance
Measured by number of iterations T to achieve accuracy ε

$$f(x_T) - f^* \leq \varepsilon$$

where $f^* = \min_x f(x)$

Algorithm Design

Algorithm Design

Idea: Use simple geometric description of the function around its minimizers to design algorithms

Algorithm Design

Idea: Use simple geometric description of the function around its minimizers to design algorithms

In this part

1. Take advantage of the sharpness of a function to accelerate convergence of classical algorithms
   Sharpness, Restart and Acceleration, V. Roulet and A. d'Aspremont, to appear in *Advances in Neural Information Processing Systems 30 (NIPS 2017).*
2. Use sharpness description to link optimization and statistical performances of decoding procedures
   Computational Complexity versus Statistical Performance on Sparse Recovery Problems, V. Roulet, N. Boumal and A. d'Aspremont, under submission to *Information and Inference: A Journal of the IMA.*

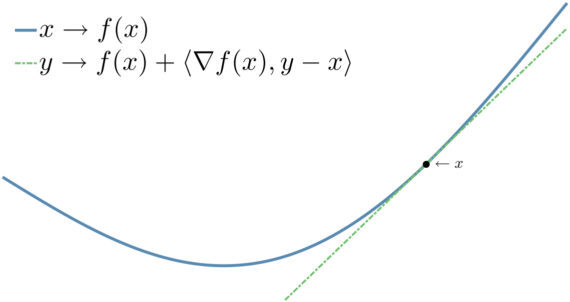
Outline

Convex Functions

Convex Functions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if at any $x \in \mathbb{R}^d$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \quad \text{for every } y \in \mathbb{R}^d$$

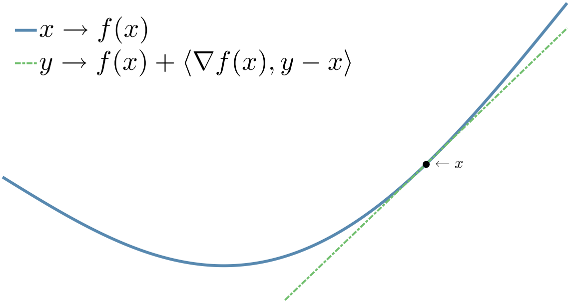


Convex Functions

Convex Functions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if at any $x \in \mathbb{R}^d$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \quad \text{for every } y \in \mathbb{R}^d$$



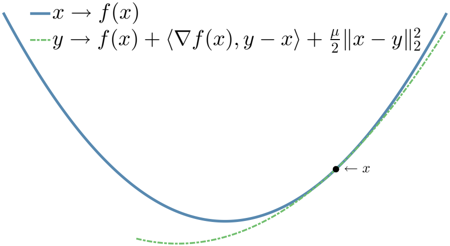
$$\nabla f(x) = 0 \implies f(x) = f^*$$

Strong Convexity

Strong Convexity

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is strongly convex if there exists $\mu \geq 0$ such that at any $x \in \mathbb{R}^d$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|x - y\|_2^2, \quad \text{for every } y \in \textbf{dom}\, f.$$

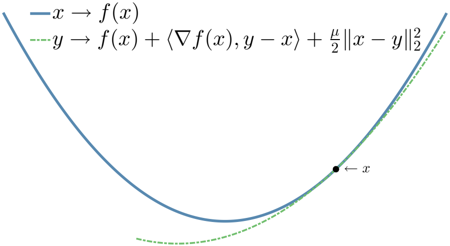


Strong Convexity

Strong Convexity

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is strongly convex if there exists $\mu \geq 0$ such that at any $x \in \mathbb{R}^d$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|x - y\|_2^2, \quad \text{for every } y \in \textbf{dom}\, f.$$



Implies minimizer x^* is unique and $\frac{\mu}{2}\|x^* - x\|_2^2 \leq f(x) - f^*$

Smooth Functions

Smooth Functions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is smooth if there exists L such that

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \text{for every } x, y \in \mathbb{R}^d$$

Smooth Functions

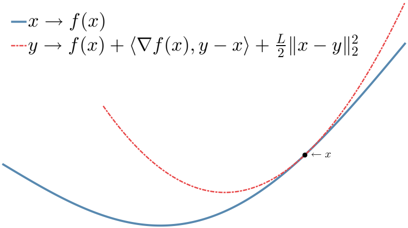
Smooth Functions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is smooth if there exists L such that

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \text{for every } x, y \in \mathbb{R}^d$$

Using Taylor expansion of f, at any point $x \in \mathbb{R}^d$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|_2^2, \quad \text{for every } y \in \mathbb{R}^d$$



Smooth Functions

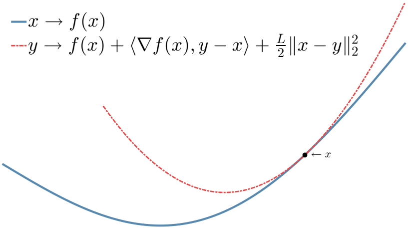
Smooth Functions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is smooth if there exists L such that

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \text{for every } x, y \in \mathbb{R}^d$$

Using Taylor expansion of f, at any point $x \in \mathbb{R}^d$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|_2^2, \quad \text{for every } y \in \mathbb{R}^d$$



$$x_{t+1} = \operatorname{argmin}_y f(x_t) + \langle \nabla f(x_t), y - x_t \rangle + \frac{L}{2}\|x_t - y\|_2^2$$
$$\Rightarrow f(x_{t+1}) \leq f(x_t) - \frac{L}{2}\|\nabla f(x_t)\|_2^2$$

Smooth Convex Optimization Problems

Study unconstrained problems

$$\text{minimize} \quad f(x)$$

where f is convex and L-smooth

Smooth Convex Optimization Problems

Study unconstrained problems

$$\text{minimize} \quad f(x)$$

where f is convex and L-smooth

Optimal algorithm

Accelerated gradient descent [Nesterov, 1983] that starts at x_0 and outputs after t iterations, $\hat{x} = \mathcal{A}(x_0, t)$, s.t.

$$f(\hat{x}) - f^* \leq \frac{4L}{t^2} d(x_0, X^*)^2,$$

where $d(x, X^*)$ is the Euclidean distance from x to $X^* = \mathrm{argmin}_x f(x)$

Intuition: Builds estimated sequence of f along iterates

Smooth Convex Optimization Problems

Study unconstrained problems

$$\text{minimize} \quad f(x)$$

where f is convex and L-smooth

Optimal algorithm

Accelerated gradient descent [Nesterov, 1983] that starts at x_0 and outputs after t iterations, $\hat{x} = \mathcal{A}(x_0, t)$, s.t.

$$f(\hat{x}) - f^* \leq \frac{4L}{t^2} d(x_0, X^*)^2,$$

where $d(x, X^*)$ is the Euclidean distance from x to $X^* = \operatorname{argmin}_x f(x)$

Intuition: Builds estimated sequence of f along iterates

Additional assumptions ?

- With strong convexity, optimal algorithm outputs $\hat{x}$ such that

$$f(\hat{x}) - f^* = O(\exp(-\sqrt{\mu/L}t))$$

Smooth Convex Optimization Problems

Study unconstrained problems

$$\text{minimize} \quad f(x)$$

where f is convex and L-smooth

Optimal algorithm

Accelerated gradient descent [Nesterov, 1983] that starts at x_0 and outputs after t iterations, $\hat{x} = \mathcal{A}(x_0, t)$, s.t.

$$f(\hat{x}) - f^* \leq \frac{4L}{t^2} d(x_0, X^*)^2,$$

where $d(x, X^*)$ is the Euclidean distance from x to $X^* = \operatorname{argmin}_x f(x)$

Intuition: Builds estimated sequence of f along iterates

Additional assumptions ?

- With strong convexity, optimal algorithm outputs $\hat{x}$ such that

$$f(\hat{x}) - f^* = O(\exp(-\sqrt{\mu/L}t))$$

- Weaker assumption ?

Outline

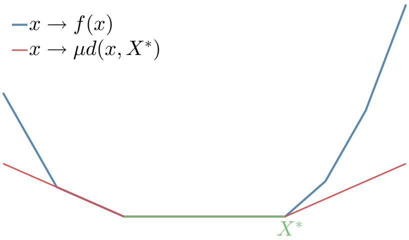
Sharpness

Sharpness

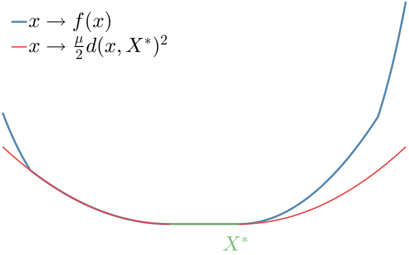
A function f satisfies the sharpness property on a set $K \supset X^*$ if there exists $r \geq 1$, $\mu > 0$, s.t.

$$\frac{\mu}{r} d(x, X^*)^r \leq f(x) - f^*, \quad \text{for every } x \in K$$

Lower bound on the function around minimizers



$r = 1$



$r = 2$

Sharpness

Applications

- Strongly convex functions ($r = 2$)
- Sparse Prediction problem like $f(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$ ($r = 2$)
- Matrix game problems $\min_x \max_y x^T Ay$ ($r = 1$)
- Real and subanalytic functions ($r = ?$)

Sharpness

Applications

- Strongly convex functions ($r = 2$)
- Sparse Prediction problem like $f(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$ ($r = 2$)
- Matrix game problems $\min_x \max_y x^T Ay$ ($r = 1$)
- Real and subanalytic functions ($r = ?$)

References

- Studied by Łojasiewicz [1963] for real analytical functions
- Numerous applications e.g. [Bolte et al., 2007]: non-convex optimization, dynamical systems, concentration inequalities...

Sharpness and Smoothness

Sharpness and Smoothness

Combining sharpness lower bound and smoothness upper bound on X^*,

$$\frac{\mu}{r} d(x, X^*)^r \leq f(x) - f^* \leq \frac{L}{2} d(x, X^*)^2$$
$$\implies 0 < \frac{2\mu}{rL} \leq \frac{d(x, X^*)^2}{d(x, X^*)^r}$$

Sharpness and Smoothness

Combining sharpness lower bound and smoothness upper bound on X^*,

$$\frac{\mu}{r} d(x, X^*)^r \leq f(x) - f^* \leq \frac{L}{2} d(x, X^*)^2$$
$$\implies 0 < \frac{2\mu}{rL} \leq \frac{d(x, X^*)^2}{d(x, X^*)^r}$$

Taking $x \to X^*$, necessarily $\mathbf{2 \leq \boldsymbol{r}}$

Sharpness and Smoothness

Combining sharpness lower bound and smoothness upper bound on X^*,

$$\frac{\mu}{r} d(x, X^*)^r \leq f(x) - f^* \leq \frac{L}{2} d(x, X^*)^2$$
$$\implies 0 < \frac{2\mu}{rL} \leq \frac{d(x, X^*)^2}{d(x, X^*)^r}$$

Taking $x \to X^*$, necessarily $\mathbf{2 \leq r}$

Condition numbers

$$\tau = 1 - 2/r \in [0, 1[\quad \text{and} \quad \kappa = L/\mu^{\frac{2}{r}}$$

Outline

Scheduled Restarts

Principle Run accelerated algorithm $\mathcal{A}$, stop it, restart from last iterate

Scheduled Restarts

Principle Run accelerated algorithm $\mathcal{A}$, stop it, restart from last iterate

Here schedule restarts in advance at times t_k and build from $x_0 \in \mathbb{R}^d$

$$x_k = \mathcal{A}(x_{k-1}, t_k)$$

Scheduled Restarts

Principle Run accelerated algorithm $\mathcal{A}$, stop it, restart from last iterate

Here schedule restarts in advance at times t_k and build from $x_0 \in \mathbb{R}^d$

$$x_k = \mathcal{A}(x_{k-1}, t_k)$$

Why?
Combine convergence bound and sharpness

$$f(x_k) - f^* \leq \frac{4L}{t_k^2} d(x_{k-1}, X^*)^2 \quad \text{and} \quad \frac{\mu}{r} d(x_{k-1}, X^*)^r \leq f(x_{k-1}) - f^*$$

So

$$f(x_k) - f^* \leq \frac{c_{L,\mu,r}}{t_k^2} (f(x_{k-1}) - f^*)^{2/r}$$

Scheduled Restarts

Principle Run accelerated algorithm $\mathcal{A}$, stop it, restart from last iterate

Here schedule restarts in advance at times t_k and build from $x_0 \in \mathbb{R}^d$

$$x_k = \mathcal{A}(x_{k-1}, t_k)$$

Why?
Combine convergence bound and sharpness

$$f(x_k) - f^* \leq \frac{4L}{t_k^2} d(x_{k-1}, X^*)^2 \quad \text{and} \quad \frac{\mu}{r} d(x_{k-1}, X^*)^r \leq f(x_{k-1}) - f^*$$

So

$$f(x_k) - f^* \leq \frac{c_{L,\mu,r}}{t_k^2} (f(x_{k-1}) - f^*)^{2/r}$$

Method of analysis

1. For given $0 < \gamma < 1$, compute t_k, $f(x_k) - f^* \leq \gamma(f(x_{k-1}) - f^*)$
2. Optimize on γ to get optimal rate in terms of total number of iterations $N = \sum_{i=1}^{R} t_k$ after R restarts

Optimal Schedule

Proposition [R. and d'Aspremont, 2017]

For f convex, L-smooth and (r, μ)-sharp on a set $K \supset \{x : f(x) \leq f(x_0)\}$
Run scheduled restarts with

$$t_k = C_{\tau,\kappa} e^{\tau k}$$

Then after R restarts and $N = \sum_{i=1}^{R} t_k$ total iterations, it outputs $\hat{x}$ s.t.

$$f(\hat{x}) - f^* = O\left(\exp(-\kappa^{-1/2} N)\right) \qquad \text{when } \tau = 0$$

$$f(\hat{x}) - f^* = O\left(1/N^{2/\tau}\right) \qquad \text{when } \tau > 0$$

Recall: $\tau = 1 - 2/r, \quad \kappa = L/\mu 2/r$

Remarks

- Optimal for this class of problems [Nemirovskii and Nesterov, 1985]
- Bound continuous in τ : for $\tau \to 0$, gets bound for $\tau = 0$

Parameter-free straegy

In practice (r, μ) are unknown, adaptivity is crucial

Parameter-free straegy

In practice (r, μ) are unknown, adaptivity is crucial

Adaptive strategy (log-scale grid search)
Given a fixed budget of iterations N, search with schedules of the form

$$t_k = Ce^{\tau k}$$

- Grid on C limited by N
- Grid on C limited by continuity of the bounds in τ

Parameter-free straegy

In practice (r, μ) are unknown, adaptivity is crucial

Adaptive strategy (log-scale grid search)
Given a fixed budget of iterations N, search with schedules of the form

$$t_k = Ce^{\tau k}$$

- Grid on C limited by N
- Grid on C limited by continuity of the bounds in τ

Analysis

- Nearly-optimal bounds (up to constant factor 4)
- Cost of the grid search $\log_2(N)^2$

Universal Scheduled Restarts

Generalization

Non-smooth or Hölder smooth convex functions where there exists $1 \leq s \leq 2$ and $L > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1}, \quad \text{for every } x, y \in \textbf{dom}\, f,$$

where $\nabla f(x)$ is any subgradient of f at x if $s = 1$ (non-smooth case)

Universal Scheduled Restarts

Generalization

Non-smooth or Hölder smooth convex functions where there exists $1 \leq s \leq 2$ and $L > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1}, \quad \text{for every } x, y \in \mathbf{dom}\, f,$$

where $\nabla f(x)$ is any subgradient of f at x if $s = 1$ (non-smooth case)

Optimal rate [Nesterov, 2015] (without sharpness)

$$f(\hat{x}) - f^* \leq \frac{c_s L d(x_0, X^*)^s}{t^\rho} \qquad \text{where } \rho = 3s/2 - 1$$

Universal Scheduled Restarts

Generalization

Non-smooth or Hölder smooth convex functions where there exists $1 \leq s \leq 2$ and $L > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1}, \quad \text{for every } x, y \in \mathbf{dom}\, f,$$

where $\nabla f(x)$ is any subgradient of f at x if $s = 1$ (non-smooth case)

Optimal rate [Nesterov, 2015] (without sharpness)

$$f(\hat{x}) - f^* \leq \frac{c_s L d(x_0, X^*)^s}{t^\rho} \qquad \text{where } \rho = 3s/2 - 1$$

Proposition [R. and d'Aspremont, 2017] (Simplified)

Optimal scheduled restart with sharpness output $\hat{x}$ s.t.

$$f(\hat{x}) - f^* = O\left(\exp(-\kappa^{-s/(2\rho)} N)\right) \qquad \text{when } \tau = 0$$

$$f(\hat{x}) - f^* = O\left(1/N^{\rho/\tau}\right) \qquad \text{when } \tau > 0$$

where $\tau = 1 - s/r \in [0, 1[$, $\kappa = L^{2/s}/\mu^{\frac{2}{r}}$

Test on Classification Problems

For real data set ($n = 208$ samples, $d = 60$ features)

Compare to

- Accelerated gradient (Acc)
- Restart heuristic enforcing monotonicity of objective values (Mono)
- Adaptive restarts (Adap)

Test on Classification Problems

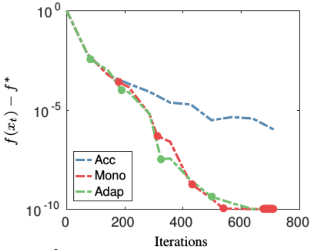
For real data set ($n = 208$ samples, $d = 60$ features)

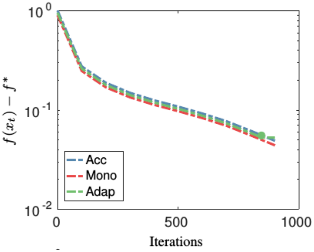
Compare to

- Accelerated gradient (Acc)
- Restart heuristic enforcing monotonicity of objective values (Mono)
- Adaptive restarts (Adap)

For

- Least square $\min_x \|Ax - b\|_2^2$
- Logistic $\min_x \sum_i \log(1 + \exp(-b_i a_i^T x))$





Least square Logistic

Large dots represent the restart iterations

Test on Classification Problems

Results also valid for composite problems (same convergence bounds)

$$\text{minimize } f(x) + g(x)$$

where f, g convex, g "simple", f is (L, s)-smooth, $f + g$ is (r, μ)-sharp

Test on Classification Problems

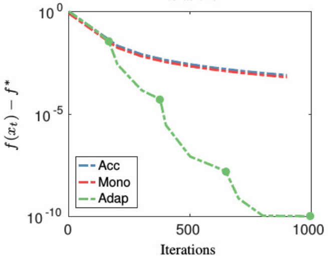
Results also valid for composite problems (same convergence bounds)

$$\text{minimize } f(x) + g(x)$$

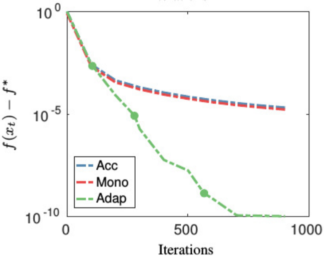
where f, g convex, g "simple", f is (L, s)-smooth, $f + g$ is (r, μ)-sharp

Test on

- Lasso $\min_x \|Ax - b\|_2^2 + \|x\|_1$
- Dual SVM $\min_x x^T AA^T x - x^T \mathbf{1} \quad \text{s.t. } 0 \leq x \leq 1$



Lasso



Dual SVM

Large dots represent the restart iterations

Outline

Sparse Recovery Problems

Sparse Recovery Problems

Goal Recover a signal $x^* \in \mathbb{R}^d$ from n linear observations

$$b_i = a_i^T x^*, \quad i \in \{1, \ldots, n\}$$

Sparse Recovery Problems

Goal Recover a signal $x^* \in \mathbb{R}^d$ from n linear observations

$$b_i = a_i^T x^*, \quad i \in \{1, \ldots, n\}$$

Problem
Needs more observations n than features d

Sparse Recovery Problems

Goal Recover a signal $x^* \in \mathbb{R}^d$ from n linear observations

$$b_i = a_i^T x^*, \quad i \in \{1, \ldots, n\}$$

Problem
Needs more observations n than features d

Additional assumption
x^* is s-sparse : has only $s \ll d$ non-zero values

Sparse Recovery Problems

Goal Recover a signal $x^* \in \mathbb{R}^d$ from n linear observations

$$b_i = a_i^T x^*, \quad i \in \{1, \ldots, n\}$$

Problem
Needs more observations n than features d

Additional assumption
x^* is s-sparse : has only $s \ll d$ non-zero values

Applications

- Coding/Decoding audio signals, images, ...
- Find explanatory variables for an experiment

Sparse Recovery Problems

Original decoding procedure
Given $b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$ and $A = (a_1, \ldots, a_n)^T \in \mathbb{R}^{n \times d}$, original problem is

$$\begin{array}{ll} \text{minimize} & \|x\|_0 \\ \text{subject to} & Ax = b \end{array}$$

where $\mathrm{Supp}(x) = \{i \in \{1, \ldots, d\}, x_i \neq 0\}$ is the support of x

Sparse Recovery Problems

Original decoding procedure
Given $b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$ and $A = (a_1, \ldots, a_n)^T \in \mathbb{R}^{n \times d}$, original problem is

$$\begin{array}{ll} \text{minimize} & \|x\|_0 \\ \text{subject to} & Ax = b \end{array}$$

where $\mathrm{Supp}(x) = \{i \in \{1, \ldots, d\}, x_i \neq 0\}$ is the support of x
$\rightarrow$ NP hard combinatorial problem...

Sparse Recovery Problems

Original decoding procedure
Given $b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$ and $A = (a_1, \ldots, a_n)^T \in \mathbb{R}^{n \times d}$, original problem is

$$\begin{array}{ll} \text{minimize} & \|x\|_0 \\ \text{subject to} & Ax = b \end{array}$$

where $\mathrm{Supp}(x) = \{i \in \{1, \ldots, d\}, x_i \neq 0\}$ is the support of x
$\rightarrow$ NP hard combinatorial problem...

Practical decoding procedure
Solve instead convex relaxation

$$\begin{array}{ll} \text{minimize} & \textbf{Card}(\mathrm{Supp}(x)) \\ \text{subject to} & Ax = b \end{array}$$

Sparse Recovery Problems

Original decoding procedure
Given $b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$ and $A = (a_1, \ldots, a_n)^T \in \mathbb{R}^{n \times d}$, original problem is

$$\begin{array}{ll} \text{minimize} & \|x\|_0 \\ \text{subject to} & Ax = b \end{array}$$

where $\mathrm{Supp}(x) = \{i \in \{1, \ldots, d\}, x_i \neq 0\}$ is the support of x
$\rightarrow$ NP hard combinatorial problem...

Practical decoding procedure
Solve instead convex relaxation

$$\begin{array}{ll} \text{minimize} & \mathbf{Card}(\mathrm{Supp}(x)) \\ \text{subject to} & Ax = b \end{array}$$

Recovery achieved if solution $\hat{x} = x^*$, where x^* original vector $(b = Ax^*)$

Sharpness in Sparse Recovery Problems

Exploit sharpness of $f(x) = \|x\|_1$ on $\{x : Ax = b\}$

$$\gamma_A d(x, X^*) \leq f(x) - f^*$$

Sharpness in Sparse Recovery Problems

Exploit sharpness of $f(x) = \|x\|_1$ on $\{x : Ax = b\}$

$$\gamma_A d(x, X^*) \leq f(x) - f^*$$

Proposition [R., Boumal and d'Aspremont, 2017] (Simplified)

Optimal scheduled restart of classical algorithm for exact recovery outputs, after N total number of iterations, $\hat{x}$ such that

$$f(\hat{x}) - f^* = O(\exp(-\gamma_A N))$$

Sharpness in Sparse Recovery Problems

Exploit sharpness of $f(x) = \|x\|_1$ on $\{x : Ax = b\}$

$$\gamma_A d(x, X^*) \le f(x) - f^*$$

Proposition [R., Boumal and d'Aspremont, 2017] (Simplified)

Optimal scheduled restart of classical algorithm for exact recovery outputs, after N total number of iterations, $\hat{x}$ such that

$$f(\hat{x}) - f^* = O(\exp(-\gamma_A N))$$

Remark

- Optimal schedule needs γ_A but log-scale grid search nearly optimal

Sharpness and Sparse Recovery Performance

Recovery threshold

Given $A \in \mathbb{R}^{n \times d}$, denote $s_{\max}(A)$ its recovery threshold such that any original signal x^* s-sparse, with $s < s_{\max}(A)$, is the unique solution of

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = Ax^* \end{array}$$

Sharpness and Sparse Recovery Performance

Recovery threshold

Given $A \in \mathbb{R}^{n \times d}$, denote $s_{\max}(A)$ its recovery threshold such that any original signal x^* s-sparse, with $s < s_{\max}(A)$, is the unique solution of

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = Ax^* \end{array}$$

Proposition [R., Boumal and d'Aspremont, 2017]

Given $A \in \mathbb{R}^{n \times d}$ and an original signal x^* s-sparse, with $s < s_{\max}(A)$,

$$\|x\|_1 - \|x^*\|_1 > (1 - \sqrt{s/s_{\max}(A)})\|x - x^*\|_1 \quad \forall x \in \mathbb{R}^d : Ax = Ax^*, x \neq x^*$$

Sharpness and Sparse Recovery Performance

Recovery threshold

Given $A \in \mathbb{R}^{n \times d}$, denote $s_{\max}(A)$ its recovery threshold such that any original signal x^* s-sparse, with $s < s_{\max}(A)$, is the unique solution of

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = Ax^* \end{array}$$

Proposition [R., Boumal and d'Aspremont, 2017]

Given $A \in \mathbb{R}^{n \times d}$ and an original signal x^* s-sparse, with $s < s_{\max}(A)$,

$$\|x\|_1 - \|x^*\|_1 > (1 - \sqrt{s/s_{\max}(A)})\|x - x^*\|_1 \quad \forall x \in \mathbb{R}^d : Ax = Ax^*, x \neq x^*$$

Rate of convergence of optimal restart scheme reads

$$\|\hat{x}\|_1 - \|x^*\|_1 = O\left(\exp\left(-\left(1 - \sqrt{s/s_{\max}(A)}\right) N\right)\right)$$

Numerical Illustration

For random observation matrix A, $s_{\max}(A) \approx n/\log d$

So to recover s-sparse signals, needs

$$n \approx s \log d$$

Numerical Illustration

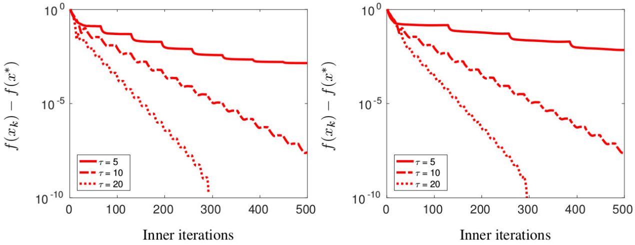
For random observation matrix A, $s_{\max}(A) \approx n/\log d$

So to recover s-sparse signals, needs

$$n \approx s \log d$$

Convergence rate of optimal restart

$$\|\hat{x}\|_1 - \|x^*\|_1 = O\left(\exp\left(-\left(1 - c\sqrt{s \log d/n}\right) N\right)\right)$$



Best restart scheme found by grid search along oversampling ratio $\tau = n/(s \log d)$ for fixed $d = 1000$

Left : sparsity $s = 20$ fixed. Right : nb of samples $n = 200$ fixed.

Conclusion and Future Work

Contributions

- Analyze acceleration of accelerated schemes by restart under a sharpness assumption
- Show cost of adaptive schemes
- Link optimization complexity and statistical performance of sparse recovery problems with sharpness

Conclusion and Future Work

Contributions

- Analyze acceleration of accelerated schemes by restart under a sharpness assumption
- Show cost of adaptive schemes
- Link optimization complexity and statistical performance of sparse recovery problems with sharpness

Not presented

- Link sharpness to robust recovery performance or noisy observations
- Extension to other sparse structures : group sparsity, low rank matrices
- Optimization algorithms seen as integration methods of the gradient flow [Scieur, R., Bach and d'Aspremont, 2017]

Conclusion and Future Work

Contributions

- Analyze acceleration of accelerated schemes by restart under a sharpness assumption
- Show cost of adaptive schemes
- Link optimization complexity and statistical performance of sparse recovery problems with sharpness

Not presented

- Link sharpness to robust recovery performance or noisy observations
- Extension to other sparse structures : group sparsity, low rank matrices
- Optimization algorithms seen as integration methods of the gradient flow [Scieur, R., Bach and d'Aspremont, 2017]

Perspectives

- Get more practical adaptive scheme.
  $\rightarrow$ Fercoq and Qu [2017] has one for $r = 2$, extends results
- Refine sharpness analysis for robust sparse recovery problems

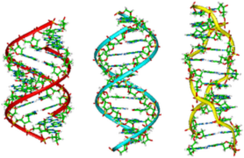
Part 2

Machine Learning Problems with Partitioning Structure

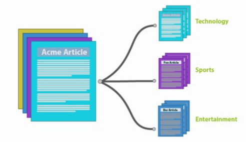
Machine Learning Problems

Goal Predict attributes y from objects x

Images → Digits



DNA → Phenotype



Documents → Topic

Supervised Machine Learning Problems

Goal

Learn mapping

$$f : x \rightarrow y$$

from n training samples of objects/attributes $(x_1, y_1), \ldots, (x_n, y_n)$

Supervised Machine Learning Problems

Goal
Learn mapping

$$f : x \to y$$

from n training samples of objects/attributes $(x_1, y_1), \ldots, (x_n, y_n)$

Method
Prediction of f on (x, y) measured by loss function $\ell(y, f(x))$
Learning procedure consist in

$$\text{minimize } \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) + R(f)$$

in mapping $f \in \mathcal{F}$ where R is a regularizer that prevents overfitting on training data

Structure Information

Idea Impose an underlying structure on the data
to both learn and simplify prediction task

Structure Information

Idea Impose an underlying structure on the data
to both learn and simplify prediction task

In this part

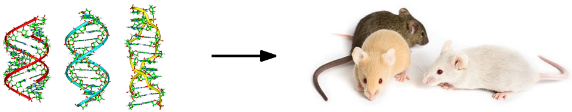
1. Group features for prediction task
   Iterative Hard Clustering of Features, V. Roulet, F. Fogel, F. Bach and A. d'Aspremont, under submission to the *21st International Conference on Artificial Intelligence and Statistics (AISTATS 2018)*

2. (Not presented) Extension to group samples or tasks
   Learning with Clustering Penalties, V. Roulet, F. Fogel, F. Bach and A. d'Aspremont, presented at workshop *Transfer and Multi-Task Learning: Trends and New Perspectives (NIPS 2015)*

Outline

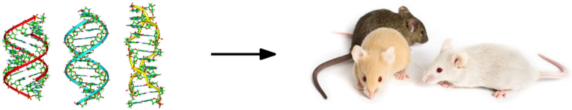
Practical Motivation

Goal Predict phenotypes from DNA



Practical Motivation

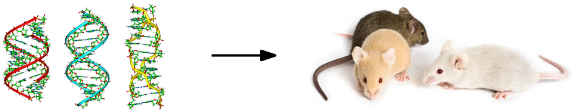
Goal Predict phenotypes from DNA



Data
Genomes $x_1, \dots x_n$ composed of d genes

Practical Motivation

Goal Predict phenotypes from DNA

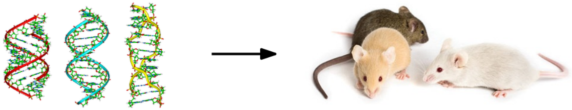


Data
Genomes $x_1, \dots x_n$ composed of d genes

Problem
Number of genes d very large, prediction hard to make and interpret

Practical Motivation

Goal Predict phenotypes from DNA



Data
Genomes $x_1, \ldots x_n$ composed of d genes

Problem
Number of genes d very large, prediction hard to make and interpret

Assumption
Some genes are redundant $\rightarrow$ form groups of genes and compute their influence

Problem Formulation

Linear regression model
Attributes $y \in \mathbb{R}$, find $w \in \mathbb{R}^d$ such that

$$x^T w \approx y$$

To this end,

$$\text{minimize } L(w) + \lambda R(w)$$

in $w \in \mathbb{R}^d$ where $L(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, w^T x_i)$ is the empirical loss and λ is a regularization parameter

Problem Formulation

Linear regression model
Attributes $y \in \mathbb{R}$, find $w \in \mathbb{R}^d$ such that

$$x^T w \approx y$$

To this end,

$$\text{minimize } L(w) + \lambda R(w)$$

in $w \in \mathbb{R}^d$ where $L(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, w^T x_i)$ is the empirical loss and λ is a regularization parameter

Examples:

- Squared loss $\ell(y, x^T w) = (y - x^T w)^2$
- Squared regularizer $R(w) = \|w\|_2^2$

Problem Formulation

Usual sparsity

Selects s features, by constraining w with at most s non-zero values,

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \textbf{Card}(\mathrm{Supp}(w)) \leq s, \end{array}$$

where $\mathrm{Supp}(w) = \{i \in \{1, \ldots, d\}, w_i \neq 0\}$ is the support of w

Problem Formulation

Usual sparsity
Selects s features, by constraining w with at most s non-zero values,

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \textbf{Card}(\mathrm{Supp}(w)) \leq s, \end{array}$$

where $\mathrm{Supp}(w) = \{i \in \{1, \dots, d\}, w_i \neq 0\}$ is the support of w
$\rightarrow$ Hard combinatorial problem

Problem Formulation

Usual sparsity

Selects s features, by constraining w with at most s non-zero values,

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \textbf{Card}(\mathrm{Supp}(w)) \leq s, \end{array}$$

where $\mathrm{Supp}(w) = \{i \in \{1, \ldots, d\}, w_i \neq 0\}$ is the support of w
$\rightarrow$ Hard combinatorial problem

$\rightarrow$ Solved approximately by projected gradient descent/convex relaxation

Problem Formulation

Usual sparsity
Selects s features, by constraining w with at most s non-zero values,

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \mathbf{Card}(\mathrm{Supp}(w)) \leq s, \end{array}$$

where $\mathrm{Supp}(w) = \{i \in \{1, \ldots, d\}, w_i \neq 0\}$ is the support of w
$\rightarrow$ Hard combinatorial problem

$\rightarrow$ Solved approximately by projected gradient descent/convex relaxation

Reduce prediction problem to at most s variables

Problem Formulation

Grouping constraints

- Constraint w to have at most Q *different* values $v_1, \ldots, v_Q$
- Each v_q is assigned to a group $g_q \subset \{1, \ldots, d\}$

Problem Formulation

Grouping constraints

- Constraint w to have at most Q *different* values $v_1, \ldots, v_Q$
- Each v_q is assigned to a group $g_q \subset \{1, \ldots, d\}$

Formally, w defines a partition $\{g_1, \ldots, g_Q\}$ of $\{1, \ldots, d\}$

$$\mathrm{Part}(w) = \{g \subset \{1, \ldots, d\} : (i, j) \in g \times g, \text{ iff } w_i = w_j\}$$

Problem Formulation

Grouping constraints

- Constraint w to have at most Q *different* values $v_1, \dots, v_Q$
- Each v_q is assigned to a group $g_q \subset \{1, \dots, d\}$

Formally, w defines a partition $\{g_1, \dots, g_Q\}$ of $\{1, \dots, d\}$

$$\mathrm{Part}(w) = \{g \subset \{1, \dots, d\} : (i, j) \in g \times g, \text{ iff } w_i = w_j\}$$

Example: $w = (\mathbf{7}, \mathbf{-3}, \mathbf{-3}, \mathbf{0}, \mathbf{7}) \quad \rightarrow \quad \mathrm{Part}(w) = \{\mathbf{\{1,5\}}, \mathbf{\{4\}}, \mathbf{\{2,3\}}\}$

Problem Formulation

Grouping constraints

- Constraint w to have at most Q *different* values $v_1, \ldots, v_Q$
- Each v_q is assigned to a group $g_q \subset \{1, \ldots, d\}$

Formally, w defines a partition $\{g_1, \ldots, g_Q\}$ of $\{1, \ldots, d\}$

$$\mathrm{Part}(w) = \{g \subset \{1, \ldots, d\} : (i, j) \in g \times g, \text{ iff } w_i = w_j\}$$

Example: $w = (\mathbf{7}, \mathbf{-3}, \mathbf{-3}, \mathbf{0}, \mathbf{7}) \quad \rightarrow \quad \mathrm{Part}(w) = \{\mathbf{\{1,5\}}, \mathbf{\{4\}}, \mathbf{\{2,3\}}\}$

Regression with grouping constraints of the feature reads

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \mathbf{Card}(\mathrm{Part}(w)) \leq Q \end{array}$$

Problem Formulation

Grouping constraints

- Constraint w to have at most Q *different* values $v_1, \ldots, v_Q$
- Each v_q is assigned to a group $g_q \subset \{1, \ldots, d\}$

Formally, w defines a partition $\{g_1, \ldots, g_Q\}$ of $\{1, \ldots, d\}$

$$\mathrm{Part}(w) = \{g \subset \{1, \ldots, d\} : (i, j) \in g \times g, \text{ iff } w_i = w_j\}$$

Example: $w = (\mathbf{7}, \mathbf{-3}, \mathbf{-3}, \mathbf{0}, \mathbf{7}) \quad \rightarrow \quad \mathrm{Part}(w) = \{\mathbf{\{1,5\}}, \mathbf{\{4\}}, \mathbf{\{2,3\}}\}$

Regression with grouping constraints of the feature reads

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \mathbf{Card}(\mathrm{Part}(w)) \leq Q \end{array}$$

Reduce prediction problem to at most Q variables

Outline

Projection on feasible set

Projection on feasible set

Projection of $w \in \mathbb{R}^d$ on $\{w : \mathbf{Card}(\mathrm{Part}(w)) \leq Q\}$ reads

$$\text{minimize} \sum_{q=1}^{Q} \sum_{i \in g_q} (w_i - v_q)^2,$$

in $v_1, \ldots, v_Q \in \mathbb{R}$ and $G = \{g_1, \ldots, g_Q\}$ a partition of $\{1, \ldots, d\}$

Projection on feasible set

Projection of $w \in \mathbb{R}^d$ on $\{w : \textbf{Card}(\mathrm{Part}(w)) \leq Q\}$ reads

$$\text{minimize} \sum_{q=1}^{Q} \sum_{i \in g_q} (w_i - v_q)^2,$$

in $v_1, \ldots, v_Q \in \mathbb{R}$ and $G = \{g_1, \ldots, g_Q\}$ a partition of $\{1, \ldots, d\}$

$\rightarrow$ Recognizes k-means in one dimension

Projection on feasible set

Projection of $w \in \mathbb{R}^d$ on $\{w : \mathbf{Card}(\mathrm{Part}(w)) \leq Q\}$ reads

$$\text{minimize} \sum_{q=1}^{Q} \sum_{i \in g_q} (w_i - v_q)^2,$$

in $v_1, \ldots, v_Q \in \mathbb{R}$ and $G = \{g_1, \ldots, g_Q\}$ a partition of $\{1, \ldots, d\}$

$\rightarrow$ Recognizes k-means in one dimension

$\rightarrow$ Can be solved exactly in polynomial time by dynamic programming

Projection on feasible set

Projection of $w \in \mathbb{R}^d$ on $\{w : \textbf{Card}(\mathrm{Part}(w)) \leq Q\}$ reads

$$\text{minimize} \sum_{q=1}^{Q} \sum_{i \in g_q} (w_i - v_q)^2,$$

in $v_1, \ldots, v_Q \in \mathbb{R}$ and $G = \{g_1, \ldots, g_Q\}$ a partition of $\{1, \ldots, d\}$

$\rightarrow$ Recognizes k-means in one dimension

$\rightarrow$ Can be solved exactly in polynomial time by dynamic programming

Projected gradient descent is possible

Iterative Hard Clustering

Denote k-means(w, Q) the projection of w that group it in Q groups

Algorithm Iterative Hard Clustering (IHC)

Inputs: $L(w), R(w), Q$, $\lambda \geq 0$, step size γ_t
Initialize $w_0 \in \mathbb{R}^d$ (e.g. $w_0 = 0$)
for t = 0,. . . ,T **do**
  $w_{t+1/2} = w_t - \gamma_t(\nabla L(w_t) + \lambda \nabla R(w_t))$
  $w_{t+1} = \text{k-means}(w_{t+1/2}, Q)$
end for
Output: $\hat{w} = w_T$

Iterative Hard Clustering

Denote k-means(w, Q) the projection of w that group it in Q groups

---

Algorithm Iterative Hard Clustering (IHC)

Inputs: $L(w), R(w), Q$, $\lambda \geq 0$, step size γ_t
Initialize $w_0 \in \mathbb{R}^d$ (e.g. $w_0 = 0$)
for t = 0,. . . ,T **do**
  $w_{t+1/2} = w_t - \gamma_t(\nabla L(w_t) + \lambda \nabla R(w_t))$
  $w_{t+1} =$ k-means$(w_{t+1/2}, Q)$
end for
Output: $\hat{w} = w_T$

Remarks

- In practice, backtracking line search for γ_t, ensures decrease
- Akin to projected gradient descent for sparse problems, called Iterative Hard Thresholding (IHT)

Convergence analysis

Convergence analysis

Problem

Constraint set is not convex, convergence is not ensured...

Convergence analysis

Problem

Constraint set is not convex, convergence is not ensured...

$\rightarrow$ Use that it's a union of subspaces defined by partitions

Convergence analysis

Problem

Constraint set is not convex, convergence is not ensured...

$\rightarrow$ Use that it's a union of subspaces defined by partitions

Recovery analysis

Analyze algorithm as a decoding procedure

- Assume
$$y_i = x_i^T w^* + \eta_i, \quad \text{for every } i \in \{1, \dots, n\}$$
where $\eta = \mathcal{N}(0, \sigma^2)$ and w^* s.t. $\mathbf{Card}(\mathrm{Part}(w^*)) \leq Q$
- Analyze convergence to w^* by solving least-square problem
- Compute number of random observations n needed to find w^*

Convergence analysis

Problem

Constraint set is not convex, convergence is not ensured...

$\rightarrow$ Use that it's a union of subspaces defined by partitions

Recovery analysis

Analyze algorithm as a decoding procedure

- Assume
$$y_i = x_i^T w^* + \eta_i, \quad \text{for every } i \in \{1, \dots, n\}$$
where $\eta = \mathcal{N}(0, \sigma^2)$ and w^* s.t. $\textbf{Card}(\mathrm{Part}(w^*)) \leq Q$
- Analyze convergence to w^* by solving least-square problem
- Compute number of random observations n needed to find w^*

Results [R., Fogel, d'Aspremont and Bach, 2017]

Recovery (up to statistical precision) for $n \geq d$ random observations

Convergence analysis

Problem

Constraint set is not convex, convergence is not ensured...

$\rightarrow$ Use that it's a union of subspaces defined by partitions

Recovery analysis

Analyze algorithm as a decoding procedure

- Assume
$$y_i = x_i^T w^* + \eta_i, \quad \text{for every } i \in \{1, \ldots, n\}$$
where $\eta = \mathcal{N}(0, \sigma^2)$ and w^* s.t. $\textbf{Card}(\mathrm{Part}(w^*)) \leq Q$
- Analyze convergence to w^* by solving least-square problem
- Compute number of random observations n needed to find w^*

Results [R., Fogel, d'Aspremont and Bach, 2017]

Recovery (up to statistical precision) for $n \geq d$ random observations

Partitioning structure much harder than sparsity where recovery needs
$$n = O(s \log d)$$

Outline

Sparse and Grouped Linear Models

Goal

Both selects s features and group them in Q groups by solving

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \mathbf{Card}(\mathrm{Supp}(w)) \leq s, \quad \mathbf{Card}(\mathrm{Part}(w)) \leq Q + 1 \end{array}$$

Sparse and Grouped Linear Models

Goal

Both selects s features and group them in Q groups by solving

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \textbf{Card}(\mathrm{Supp}(w)) \leq s, \quad \textbf{Card}(\mathrm{Part}(w)) \leq Q + 1 \end{array}$$

Procedure

- Develop new dynamic programming to project on constraints
- Use resulting projected gradient descent

Sparse and Grouped Linear Models

Goal
Both selects s features and group them in Q groups by solving

$$\begin{array}{ll} \text{minimize} & L(w) + \lambda R(w) \\ \text{subject to} & \textbf{Card}(\mathrm{Supp}(w)) \leq s, \quad \textbf{Card}(\mathrm{Part}(w)) \leq Q + 1 \end{array}$$

Procedure

- Develop new dynamic programming to project on constraints
- Use resulting projected gradient descent

Recovery analysis
Constraint set is still a union of subspaces, same analysis applied
But here

$$n = O(s \log d + Q \log s + (s - Q) \log Q)$$

observations are sufficient

Outline

Synthetic Experiments

Setting

- $y_i = x_i^T w^* + \eta_i$ with $\eta \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
- w^* composed of $Q = 5$ group of identical features among $d = 100$

Synthetic Experiments

Setting

- $y_i = x_i^T w^* + \eta_i$ with $\eta \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
- w^* composed of $Q = 5$ group of identical features among $d = 100$

Goal

- Test robustness of our method with n and level of noise σ
- Measure $\|w_* - \hat{w}\|_2$ with $\hat{w}$ estimated vector

Synthetic Experiments Results

Compare Iterative Hard Clustering (IHC) to

- Least square given original partition $\mathrm{Part}(w^*)$ (Oracle)
- Least-squares (LS)
- Least-squares followed by a k-means (LSK)
- OSCAR penalty (enforces cluster with regularization) (OS)

Synthetic Experiments Results

Compare Iterative Hard Clustering (IHC) to

- Least square given original partition $\mathrm{Part}(w^*)$ (Oracle)
- Least-squares (LS)
- Least-squares followed by a k-means (LSK)
- OSCAR penalty (enforces cluster with regularization) (OS)

	$n = 50$	$n = 75$	$n = 100$	$n = 125$	$n = 150$
Oracle	$0.16_{\pm 0.06}$	$0.14_{\pm 0.04}$	$0.10_{\pm 0.04}$	$0.10_{\pm 0.04}$	$0.09_{\pm 0.03}$
LS	$61.94_{\pm 17.63}$	$51.94_{\pm 16.01}$	$21.41_{\pm 9.40}$	$1.02_{\pm 0.18}$	$0.70_{\pm 0.09}$
LSK	$62.93_{\pm 18.05}$	$57.78_{\pm 17.03}$	$10.18_{\pm 14.96}$	$0.31_{\pm 0.19}$	$0.19_{\pm 0.12}$
OS	$61.54_{\pm 17.59}$	$52.87_{\pm 15.90}$	$11.32_{\pm 7.03}$	$1.25_{\pm 0.28}$	$0.71_{\pm 0.10}$
IHC	$63.31_{\pm 18.24}$	$52.72_{\pm 16.51}$	$5.52_{\pm 14.33}$	$0.14_{\pm 0.09}$	$0.09_{\pm 0.04}$

Measure of $\|w_* - \hat{w}\|_2$ along number of samples n for fixed $\sigma = 0.5, d = 100$

Synthetic Experiments Results

Compare Iterative Hard Clustering (IHC) to

- Least square given original partition $\mathrm{Part}(w^*)$ (Oracle)
- Least-squares (LS)
- Least-squares followed by a k-means (LSK)
- OSCAR penalty (enforces cluster with regularization) (OS)

	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 1$
Oracle	0.86 $\pm$0.27	1.72$\pm$0.54	8.62$\pm$2.70	17.19$\pm$5.43
LS	7.04$\pm$0.92	14.05$\pm$1.82	70.39$\pm$9.20	140.41$\pm$18.20
LSK	1.44$\pm$0.46	2.88$\pm$0.91	19.10$\pm$12.13	48.09$\pm$27.46
OS	14.43$\pm$2.45	18.89$\pm$3.46	71.00$\pm$10.12	140.33$\pm$18.83
IHC	0.87$\pm$0.27	1.74$\pm$0.52	9.11$\pm$4.00	26.23$\pm$18.00

Measure of $\|w_* - \hat{w}\|_2$ along level of noise σ for fixed $n = 150, d = 100$

Conclusion and Future Work

Contributions

- Developed simple method to group (and select) features
- Analyzed recovery performance in comparison to sparsity
- Provides scalable and robust results

Conclusion and Future Work

Contributions

- Developed simple method to group (and select) features
- Analyzed recovery performance in comparison to sparsity
- Provides scalable and robust results

Not presented

- Convex approach for least square loss
- Systematic method to group features, samples or tasks

Conclusion and Future Work

Contributions

- Developed simple method to group (and select) features
- Analyzed recovery performance in comparison to sparsity
- Provides scalable and robust results

Not presented

- Convex approach for least square loss
- Systematic method to group features, samples or tasks

Perspectives

- Test on genomic data (not satisfying yet on text)
- Refine partitioning constraint to get better recovery results
- Develop regularization norm from partitioning constraints

Thank you for your attention !

References

Bolte, J., Daniilidis, A. and Lewis, A. [2007], 'The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems', *SIAM Journal on Optimization* **17**(4), 1205–1223.

Fercoq, O. and Qu, Z. [2017], 'Adaptive restart of accelerated gradient methods under local quadratic growth condition', *arXiv preprint arXiv:1709.02300* .

Łojasiewicz, S. [1963], 'Une propriété topologique des sous-ensembles analytiques réels', *Les équations aux dérivées partielles* pp. 87–89.

Nemirovskii, A. and Nesterov, Y. [1985], 'Optimal methods of smooth convex minimization', *USSR Computational Mathematics and Mathematical Physics* **25**(2), 21–30.

Nesterov, Y. [1983], 'A method of solving a convex programming problem with convergence rate $O(1/k^2)$', *Soviet Mathematics Doklady* **27**(2), 372–376.

Nesterov, Y. [2015], 'Universal gradient methods for convex optimization problems', *Mathematical Programming* **152**(1-2), 381–404.

R., V., Boumal, N. and d'Aspremont, A. [2017], 'Complexity verus statistacal performance on sparse recovery problems', *Submitted at Information and Inference : A Journal of the IMA* .

R., V. and d'Aspremont, A. [2017], 'Sharpness, restart and acceleration', *Advances in Neural Information Processing Systems 30* .

R., V., Fogel, F., d'Aspremont, A. and Bach, F. [2017], 'Iterative hard clustering of features', *Submitted at 21st International Conference on Artificial Intelligence and Statistics (AISTATS 2018)* .