

# Feature Selection and Sparse Learning

Jiayu Zhou

<sup>1</sup>Department of Computer Science and Engineering  
Michigan State University  
East Lansing, MI USA

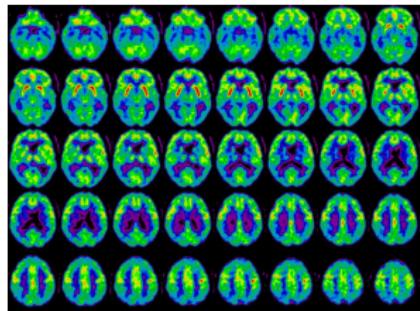
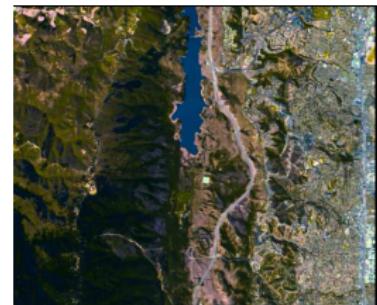
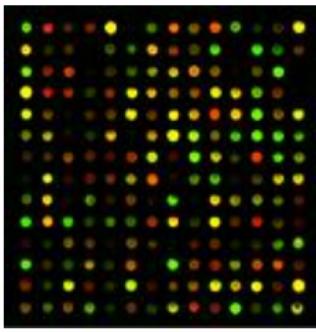
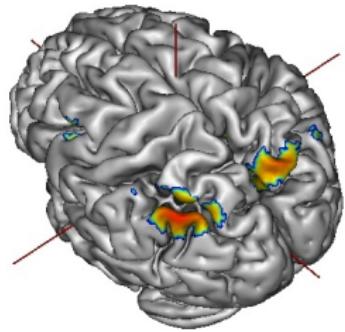
April 14, 2016

# Table of contents

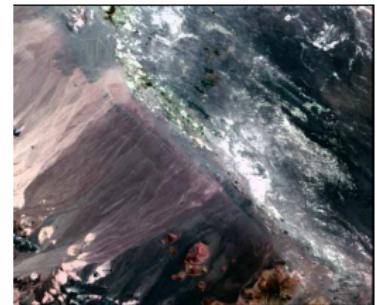
- 1 Feature Selection
- 2 Sparse Learning
- 3 Advanced Topics on Sparsity
  - Group Lasso
- 4 Algorithms
  - Greedy Algorithms
  - Optimization

# Feature Selection

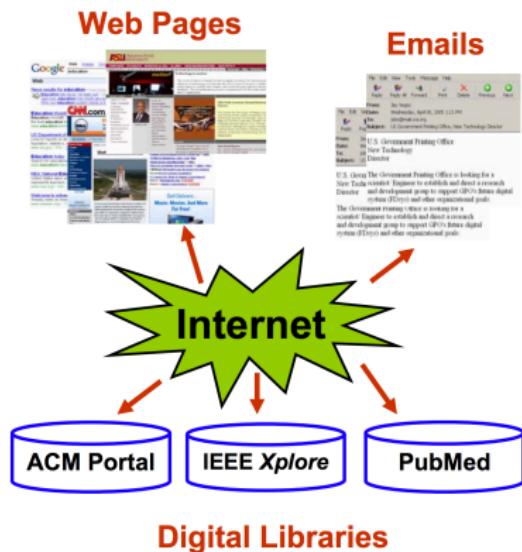
# Mining High-Dimensional Data



QFDACCFIGDDVSKIYR-DYGPI  
QFDACCFIGDDVSKIYR-DHGPI  
QFGACCFIIDDVSKTFRLHDGPI  
QFDAC-FIIDDVSKIFRLHDGPI  
RFDASCFIGDDVSKIFRLHDGPI  
QFSVYCLIIDDVSKIYR-HDGPM  
QFPVCSIIDDLSSKMYR-HDSPV  
QFFVFCLIIDDLSSKIYR-DDGLI  
QFDARCFIIDDLSKIYR-HDGQV  
QFDARCFIIDDLSKIYR-HDGPI  
RFDACCFIGDDVSKICK-HDGPV  
QFDACCFIGDDVSKICK-HDGPV



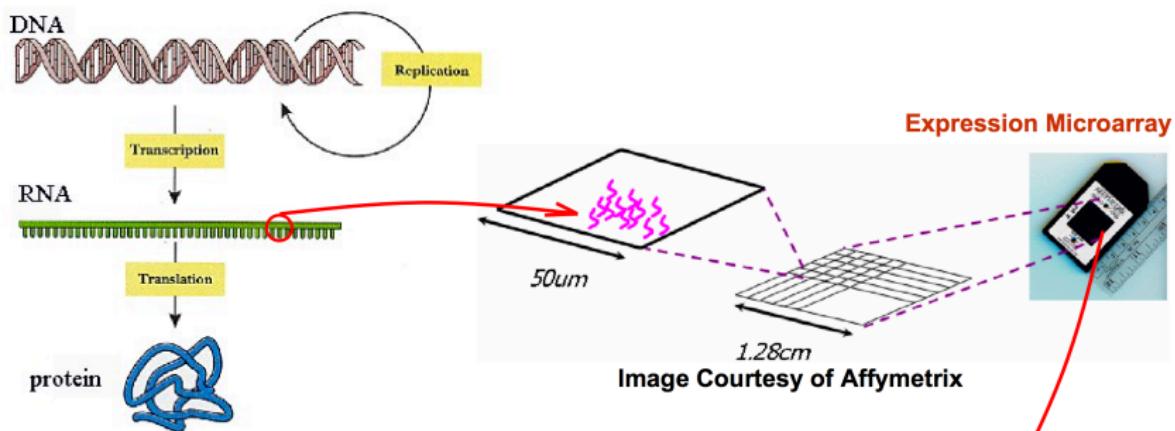
# Document Classification



	Terms				
	T <sub>1</sub>	T <sub>2</sub>	.....	T <sub>N</sub>	C
D <sub>1</sub>	12	0	.....	6	Sports
D <sub>2</sub>	3	10	.....	28	Travel
:	:			:	:
D <sub>M</sub>	0	11	.....	16	Jobs

- **Task:** To classify unlabeled documents into categories
- **Challenge:** thousands of terms
- **Solution:** to apply dimensionality reduction

# Gene Expression Microarray Analysis



- Task:** To classify novel samples into known disease types (disease diagnosis)
- Challenge:** thousands of genes, few samples
- Solution:** to apply dimensionality reduction

Gene Sample	M23197_at	U66497_at	M192287_at	...	Class
Sample 1	261	88	4778	...	ALL
Sample 2	101	74	2700	...	ALL
Sample 3	1450	34	498	...	AML
.	.	.	.	...	.
.	.	.	.	...	.
.	.	.	.	...	.

Expression Microarray Data Set

# Major Techniques of Dimensionality Reduction

## Feature Extraction

- Feature reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space
- Given a set of data points of  $p$  variables  $\{x_1, \dots, x_n\}$ , Compute their low-dimensional representation:

$$x_i \in \mathbb{R}^d \rightarrow y_i \in \mathbb{R}^p (p \ll d)$$

- Criterion for feature reduction can be different based on different problem settings
  - Unsupervised setting: minimize the information loss
  - Supervised setting: maximize the class discrimination

# Major Techniques of Dimensionality Reduction

## Feature Selection

- A process that chooses an optimal subset of features according to a objective function
- Objectives
  - To reduce dimensionality and remove noise
  - To improve mining performance
    - Speed of learning
    - Predictive accuracy
    - Simplicity and comprehensibility of mined results

# Feature Reduction vs Feature Selection

- Feature reduction
  - All original features are used
  - The transformed features are linear combinations of the original features
- Feature selection
  - Only a subset of the **original** features are selected
- Dimension Reduction Tutorial

SIAM Data Mining 2007 Tutorial (Yu, Ye, and Liu): “Dimensionality Reduction for Data Mining - Techniques, Applications, and Trends”

<http://www.cs.binghamton.edu/~lyu/SDM07/DR-SDM07.pdf>

# Models of Feature Selection

## Filter Model

- Separating feature selection from classifier learning
- Relying on general characteristics of data (information, distance, dependence, consistency)
- No bias toward any learning algorithm, fast

# Models of Feature Selection

## Filter Model

- Separating feature selection from classifier learning
- Relying on general characteristics of data (information, distance, dependence, consistency)
- No bias toward any learning algorithm, fast

## Wrapper Model

- Relying on a predetermined classification algorithm
- Using predictive accuracy as goodness measure
- High accuracy, computationally expensive

# Models of Feature Selection

## Filter Model

- Separating feature selection from classifier learning
- Relying on general characteristics of data (information, distance, dependence, consistency)
- No bias toward any learning algorithm, fast

## Wrapper Model

- Relying on a predetermined classification algorithm
- Using predictive accuracy as goodness measure
- High accuracy, computationally expensive

## Embedded Method

- Takes advantage of its own variable selection algorithm
- Combines the advantages of both previous methods
- Relies on preliminary knowledge of a good selection.

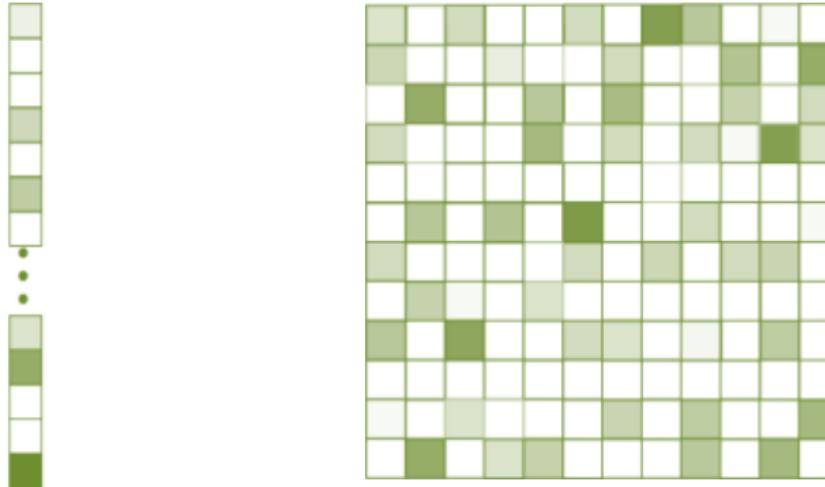
# Sparse Learning

# Why Sparse Learning

- **Embed** dimensionality reduction into data mining tasks
- Flexible models for complex feature **structures**
- Strong **theoretical** guarantee
- **Convex** formulations
- Empirical success in many **applications**
- Recent progress on **efficient** implementations

# What is Sparsity

- Many data mining tasks can be represented using a vector or a matrix.
- “Sparsity” implies many zeros in a vector or a matrix.



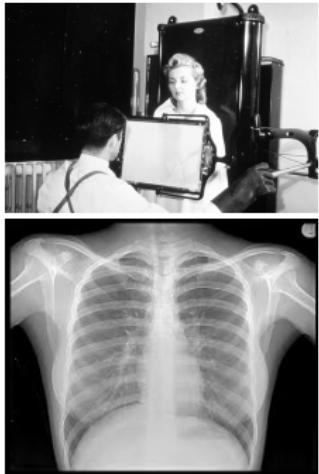
# Human Anatomy



Anatomy Lesson of Dr. Nicolaes Tulp by Rembrandt van Rijn, 1632.

# Biomedical Imaging

X-Ray, 1895



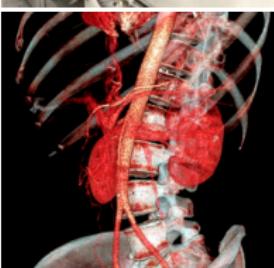
1901 Nobel Prize in  
Physics  
Wilhelm Röntgen's

# Biomedical Imaging

X-Ray, 1895



1901 Nobel Prize in  
Physics  
Wilhelm Röntgen's

Computed Tomography  
(CT), 1967

1979 Nobel Prize in  
Physiology or Medicine  
Allan M. Cormack and  
Godfrey N. Hounsfield

# Biomedical Imaging

X-Ray, 1895



Computed Tomography (CT), 1967



Magnetic Resonance Imaging (MRI), 1971

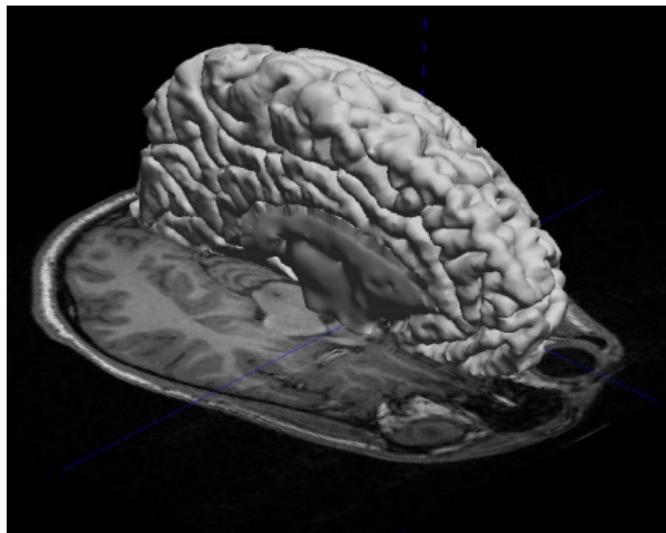


1901 Nobel Prize in Physics  
Wilhelm Röntgen's

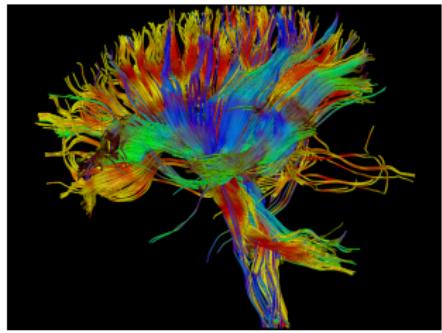
1979 Nobel Prize in Physiology or Medicine  
Allan M. Cormack and Godfrey N. Hounsfield

2003 Nobel Prize in Physiology or Medicine  
Paul Lauterbur and Sir Peter Mansfield

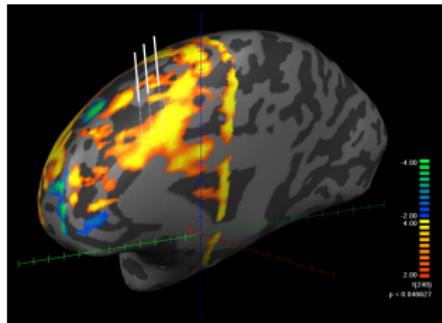
# Magnetic Resonance Imaging



Structural



Diffusion



Functional

# The Sensing Problem

- Acquire a digital object  $x \in \mathbb{R}^p$  from  $n$  measurements:

$$y_i = \langle x, \varphi_i \rangle, i = 1, 2, \dots, n$$

where waveforms  $\varphi_i$  are sinusoids,  $y$  is a vector of Fourier coefficients (e.g., measurements acquired by MRI)

- Recover the object from the measurements  
Solving a linear system of equations

# Magnetic Resonance Imaging (cont.)



# Compressive Sensing?

- Is accurate reconstruction possible from  $n \ll p$  measurements only?
  - Few sensors
  - Measurements are very expensive
  - Sensing process is slow
  - Save lives

# Compressive Sensing?

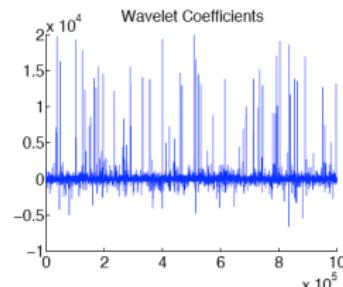
- Is accurate reconstruction possible from  $n \ll p$  measurements only?
  - Few sensors
  - Measurements are very expensive
  - Sensing process is slow
  - Save lives
- Conventional wisdom: reconstruction is impossible
  - Number of measurements must match the number of unknowns

$$\begin{matrix} y \\ | \\ n \times 1 \text{ measurements} \end{matrix} = \begin{matrix} A = [\varphi_1^T; \varphi_2^T; \dots; \varphi_n^T] \\ | \\ p \times 1 \text{ signal} \end{matrix}$$

If  $n \ll p$ , the system is underdetermined.

# Compressive Sensing Works

- Many natural signals are sparse or compressible in the sense that they have concise representations when expressed in the proper basis
- Megapixel image represented as 2.5% largest wavelet coefficients (Candes and Wakin, 2008)



# MRI by Compressive Sensing



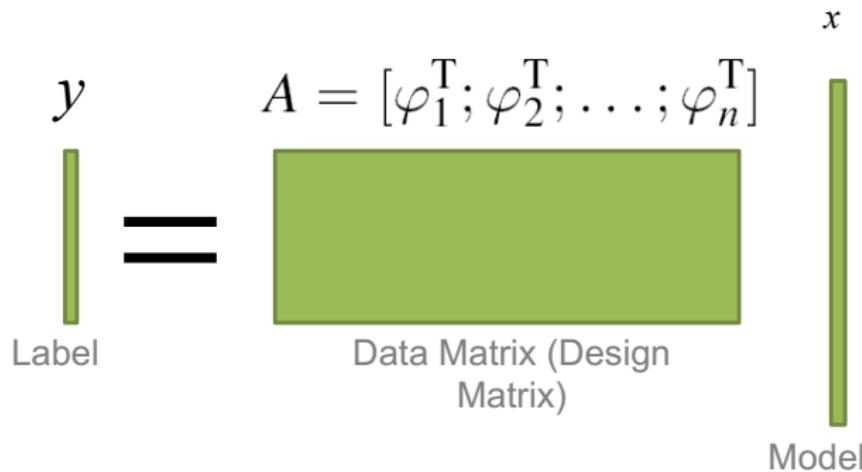
# Sparsity

Dominant modeling tool in

- Genomics and Genetics
- Signal and audio processing
- Image processing
- Neuroscience (theory of sparse coding)
- **Machine learning and data mining**
- ...

# Sparsity in Machine Learning

Regression, classification, collaborative filtering ...

$$\begin{matrix} y \\ \text{Label} \end{matrix} = \begin{matrix} A = [\varphi_1^T; \varphi_2^T; \dots; \varphi_n^T] \\ \text{Data Matrix (Design Matrix)} \end{matrix} \begin{matrix} x \\ \text{Model} \end{matrix}$$


# Convex Sparse Learning Models

- Let  $x$  be the model parameter to be estimated. A commonly employed model for estimating  $x$  is

$$\min_x \mathcal{L}(x) + \lambda \mathcal{R}(x)$$

- equivalent to the following model:

$$\min_x \mathcal{L}(x) \quad \text{s.t. } \mathcal{R}(x) \leq z$$

Theorem 1 in Efficient and Accurate  $\ell_p$ -Norm Multiple Kernel Learning, NIPS 2009

# Convex Sparse Learning Models

- Let  $x$  be the model parameter to be estimated. A commonly employed model for estimating  $x$  is:

$$\min_x \mathcal{L}(x) + \lambda \mathcal{R}(x)$$

- Sparsity via  $\ell_1$  and Elastic Net
- Sparsity via  $\ell_1/\ell_q$
- Sparsity via Fused Lasso
- Sparse Inverse Covariance Estimation
- Sparsity via Trace Norm

# Sparsity via Cardinality Minimization

- Many problems in scientific computing can be cast as

$$\min_x \text{Card}(x) \quad x \in \mathcal{P}$$

where  $\text{Card}(x)$ . denotes the cardinality (nnz) of the vector  $x$ .

- Such problems seek a “sparse” solution, one with many zeroes in it.

# Sparsity via Cardinality Minimization

- Many problems in scientific computing can be cast as

$$\min_x \text{Card}(x) \quad x \in \mathcal{P}$$

where  $\text{Card}(x)$ . denotes the cardinality (nnz) of the vector  $x$ .

- Such problems seek a “sparse” solution, one with many zeroes in it.
- A related problem is a penalized version of the above, where we seek to trade-off an objective function against cardinality:

$$\min_x \mathcal{L}(x) + \lambda \text{Card}(x)$$

# Sparsity via Cardinality Minimization

- Many problems in scientific computing can be cast as

$$\min_x \text{Card}(x) \quad x \in \mathcal{P}$$

where  $\text{Card}(x)$ . denotes the cardinality (nnz) of the vector  $x$ .

- Such problems seek a “sparse” solution, one with many zeroes in it.
- A related problem is a penalized version of the above, where we seek to trade-off an objective function against cardinality:

$$\min_x \mathcal{L}(x) + \lambda \text{Card}(x)$$

- Cardinality minimization (aka  $\ell_0$  “norm”) is a hard problem in general. It appears in many areas, such as classification.

# Sparsity via $\ell_1$ -Norm Penalty

- We use the convex envelope of  $\ell_0$  norm:  $\ell_1$  norm:

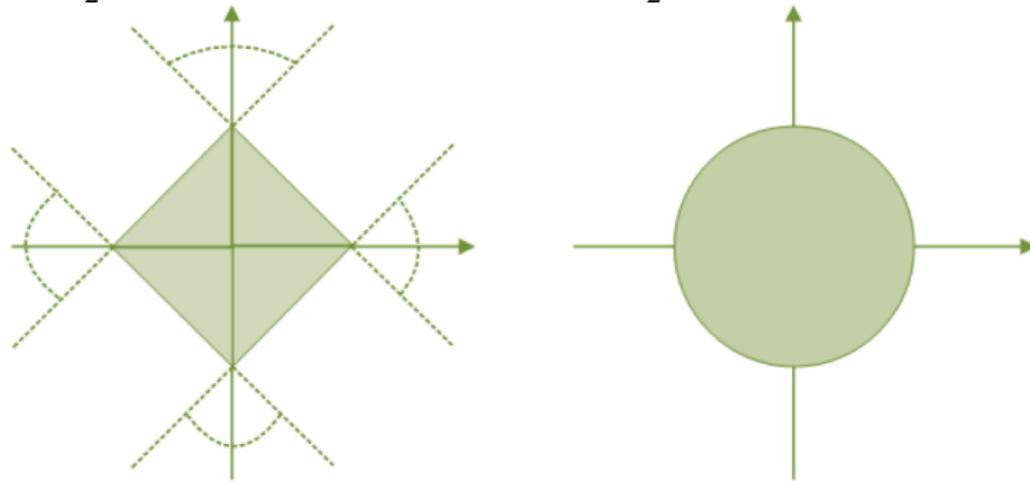
$$\min_x \mathcal{L}(x) + \lambda \|x\|_1$$

- Attracting properties
  - Sparsity induced norm
  - Valid norm
  - Convex
  - Computational tractable
  - Theoretical properties
  - Various extensions

# Why does $\ell_1$ -Norm Induce Sparsity?

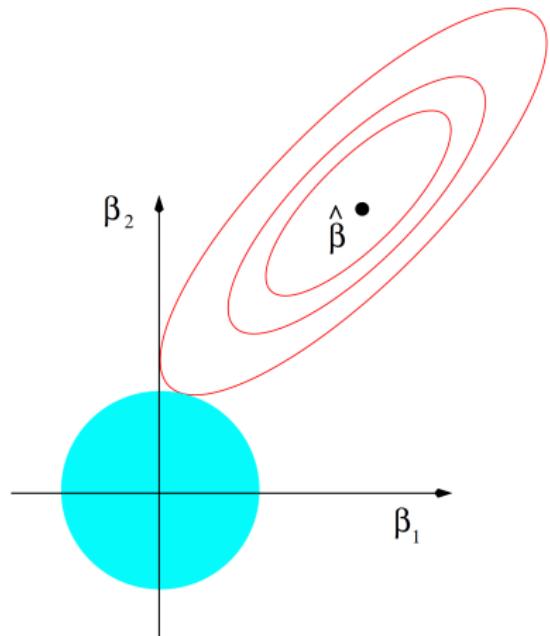
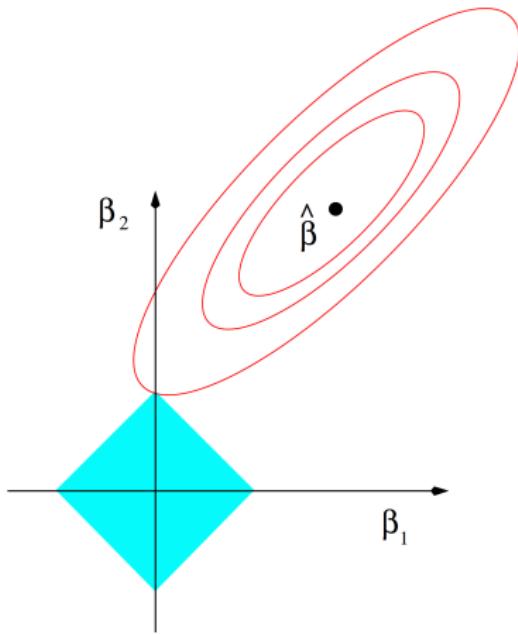
Understand from projection

$$\min_x \frac{1}{2} \|x - v\|^2 \text{ s.t. } \|x\|_1 \leq 1 \quad \min_x \frac{1}{2} \|x - v\|^2 \text{ s.t. } \|x\|_2 \leq 1$$



# Why does $\ell_1$ -Norm Induce Sparsity?

Understand from constrained optimization



# Least Squares, Ridge and Lasso Regression

Under the orthogonal design ( $A^T A = I$ )

- Least Squares  $\min_x \frac{1}{2} \|Ax - y\|_2^2$

# Least Squares, Ridge and Lasso Regression

Under the orthogonal design ( $A^T A = I$ )

- Least Squares  $\min_x \frac{1}{2} \|Ax - y\|_2^2$

$$x_{LS} = (A^T A)^{-1} A^T y = A^T y$$

# Least Squares, Ridge and Lasso Regression

Under the orthogonal design ( $A^T A = I$ )

- Least Squares  $\min_x \frac{1}{2} \|Ax - y\|_2^2$

$$x_{LS} = (A^T A)^{-1} A^T y = A^T y$$

- Ridge Regression  $\min_x \frac{1}{2} \|Ax - y\|_2^2 + \frac{\lambda}{2} \|x\|_2^2$

# Least Squares, Ridge and Lasso Regression

Under the orthogonal design ( $A^T A = I$ )

- Least Squares  $\min_x \frac{1}{2} \|Ax - y\|_2^2$

$$x_{LS} = (A^T A)^{-1} A^T y = A^T y$$

- Ridge Regression  $\min_x \frac{1}{2} \|Ax - y\|_2^2 + \frac{\lambda}{2} \|x\|_2^2$

$$x_{RR} = (A^T A + \lambda I)^{-1} A^T y = A^T y / (1 + \lambda) = x_{LS} / (1 + \lambda)$$

# Least Squares, Ridge and Lasso Regression

Under the orthogonal design ( $A^T A = I$ )

- Least Squares  $\min_x \frac{1}{2} \|Ax - y\|_2^2$

$$x_{LS} = (A^T A)^{-1} A^T y = A^T y$$

- Ridge Regression  $\min_x \frac{1}{2} \|Ax - y\|_2^2 + \frac{\lambda}{2} \|x\|_2^2$

$$x_{RR} = (A^T A + \lambda I)^{-1} A^T y = A^T y / (1 + \lambda) = x_{LS} / (1 + \lambda)$$

- Lasso Regression  $\min_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1$

# Least Squares, Ridge and Lasso Regression

Under the orthogonal design ( $A^T A = I$ )

- Least Squares  $\min_x \frac{1}{2} \|Ax - y\|_2^2$

$$x_{LS} = (A^T A)^{-1} A^T y = A^T y$$

- Ridge Regression  $\min_x \frac{1}{2} \|Ax - y\|_2^2 + \frac{\lambda}{2} \|x\|_2^2$

$$x_{RR} = (A^T A + \lambda I)^{-1} A^T y = A^T y / (1 + \lambda) = x_{LS} / (1 + \lambda)$$

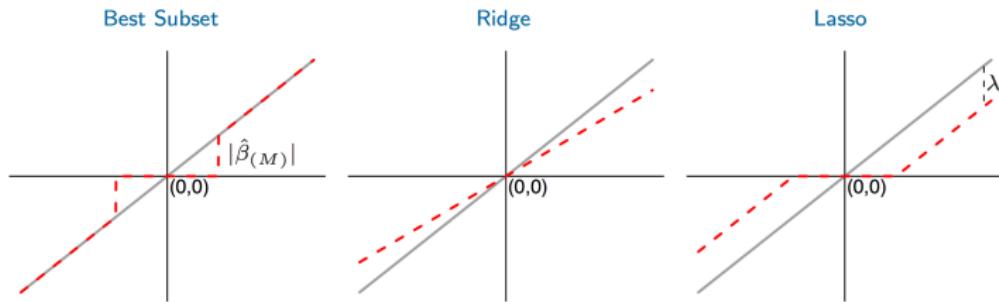
- Lasso Regression  $\min_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1$

$$x_{LA} = \frac{1}{2} \|x - A^T y\|_F^2 + \lambda \|x\|_1 = \text{sign}(x_{LS}) \max(0, |x_{LS}| - \lambda)$$

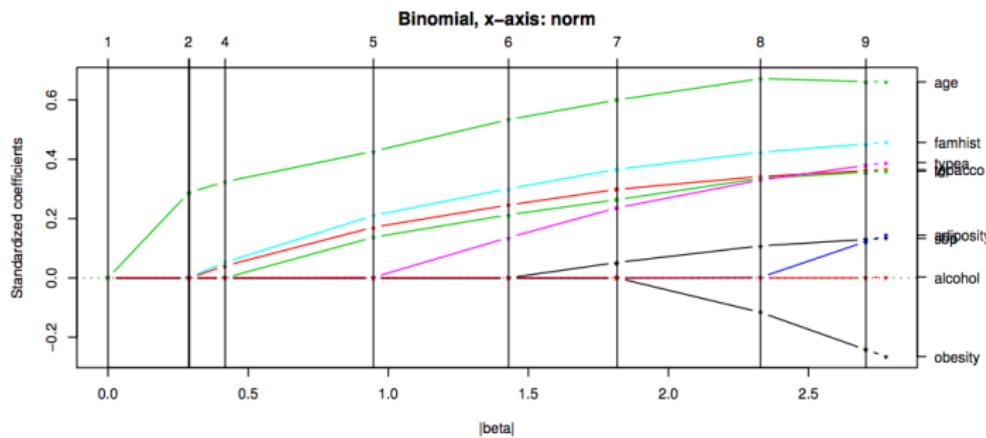
# Least Squares, Ridge and Lasso Regression

Under the orthogonal design ( $A^T A = I$ )

- Best Subset Regression  $x_{BS} = x_{LS} \cdot I(|x_{LS}| \geq |x_{(M)}|)$
- Ridge Regression  $x_{RR} = x_{LS}/(1 + \lambda)$
- Lasso Regression  $x_{LA} = \text{sign}(x_{LS}) \max(0, |x_{LS}| - \lambda)$



# Lasso Path



$\ell_1$  regularization path in a constrained Lasso problem.

# Elastic Net

## Limitation of Lasso

- When  $p > n$ , the lasso selects at most  $n$  features before it saturates
- If there is a group of highly correlated features, then the lasso tends to select one feature from a group and ignore the others
- Not stable in correlated design.

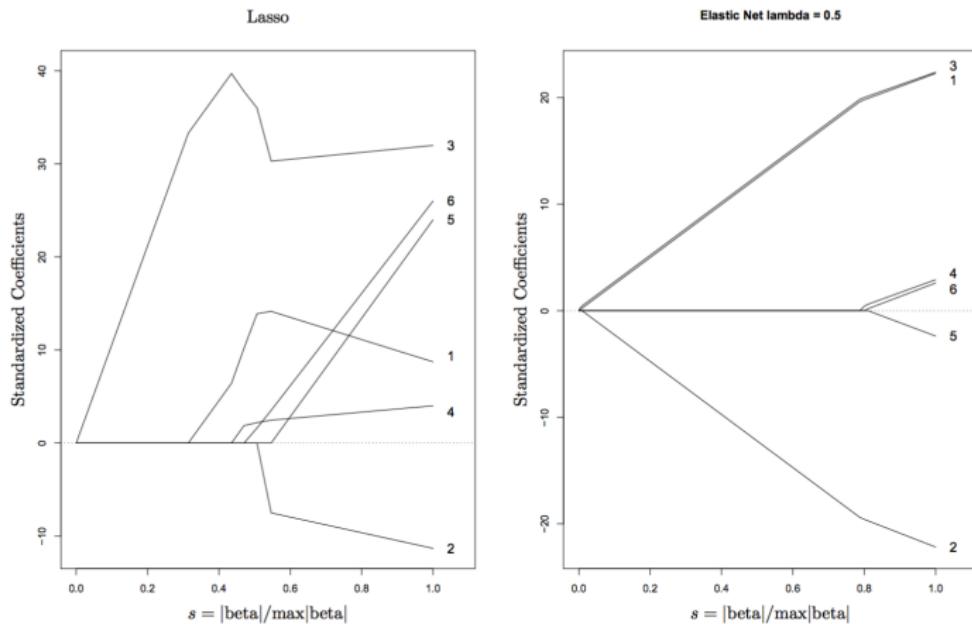
## Elastic Net

- Linearly combines  $\ell_1$  and  $\ell_2$  penalties of the lasso and ridge

$$\min_x \|Ax - y\|_2^2 + \lambda_1\|x\|_1 + \lambda_2\|x\|_2^2$$

- Removes the limitation on the number of selected variables
- Encourages *grouping* effect
- Stabilizes the  $\ell_1$  regularization path.

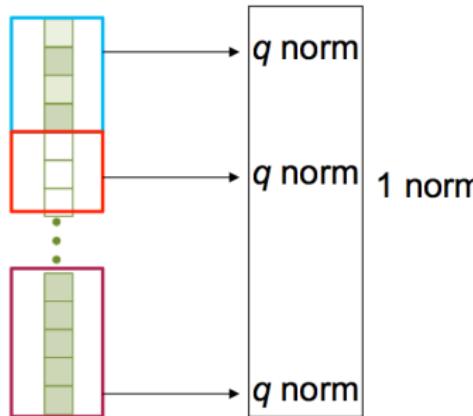
# Elastic Net Regularization Path



# Advanced Topics on Sparsity

# From $\ell_1$ to $\ell_1/\ell_q$

- Pre-defined groups within the features
  - Genes that belong to the same biological pathway
  - Collections of indicator (dummy) variables
- Let  $\mathcal{G}$  collectively denote a set of group assignments  $\{G_1, \dots, G_L\}$
- Define  $\ell_1/\ell_q$  norm as  $\|x\|_{q,1} = \sum_{l=1}^L \|x_{G_l}\|_q$ 
  - Most existing work focus on  $q = 2, \infty$
  - What about  $q = 1$ ?



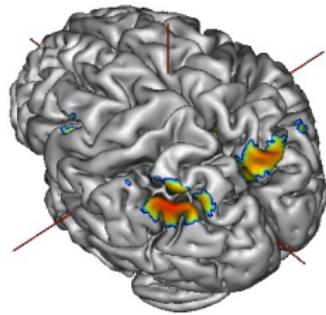
# Group Lasso

$$\begin{array}{c}
 y \\
 \parallel \\
 = \\
 A \\
 \vdots \\
 \parallel \\
 n \times 1
 \end{array}
 \quad
 \begin{array}{c}
 x^* \\
 \times \\
 \vdots \\
 \parallel \\
 p \times 1
 \end{array}
 \quad
 +
 \quad
 \begin{array}{c}
 z \\
 \parallel \\
 n \times 1
 \end{array}$$

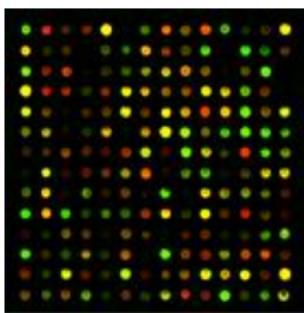
$p \gg n$

$$\min_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \sum_{l=1}^L \sqrt{|G_i|} \|x_{G_i}\|_2$$

# Applications of Group Feature Selection



brain region



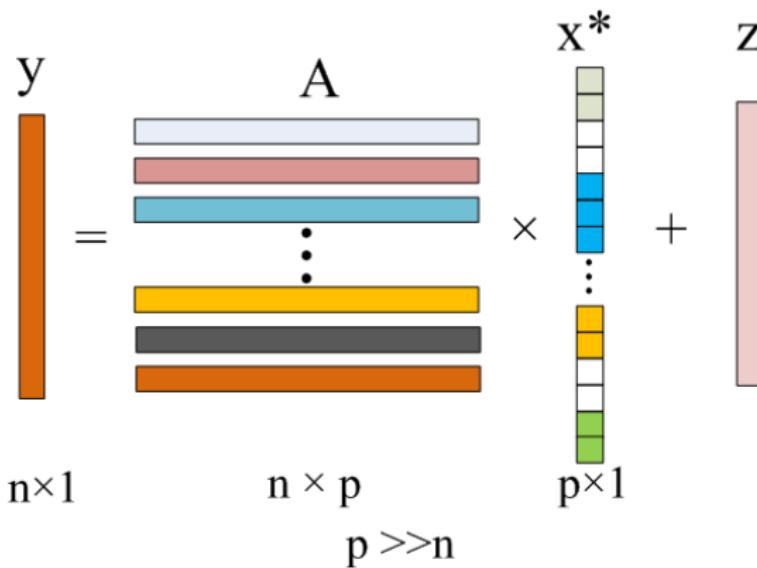
functional group

group

	group			
A	1	0	0	0
T	0	1	0	0
C	0	0	1	0
G	0	0	0	1

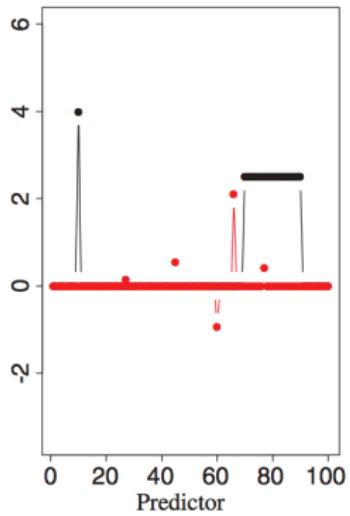
categorical variable

# Fused Lasso

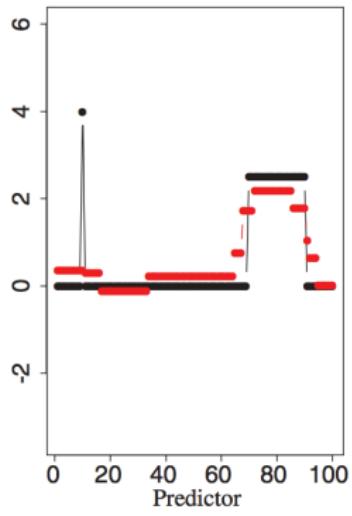


$$\min_{\mathbf{x}} \frac{1}{2} \| \mathbf{A} \mathbf{x} - \mathbf{y} \|_2^2 + \lambda_1 \sum_{i=1}^p |x_i| + \lambda_2 \sum_{i=1}^{p-1} |x_i - x_{i+1}|$$

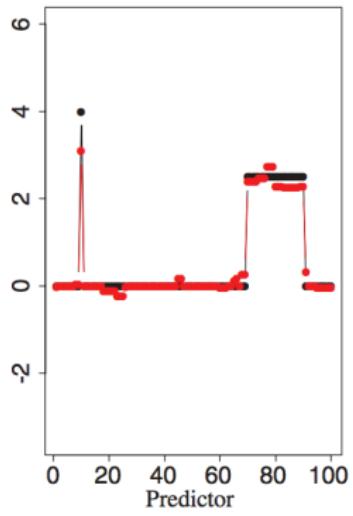
# Fused Lasso Experiment



(a)



(b)



(c)

# Many other extensions of sparsity

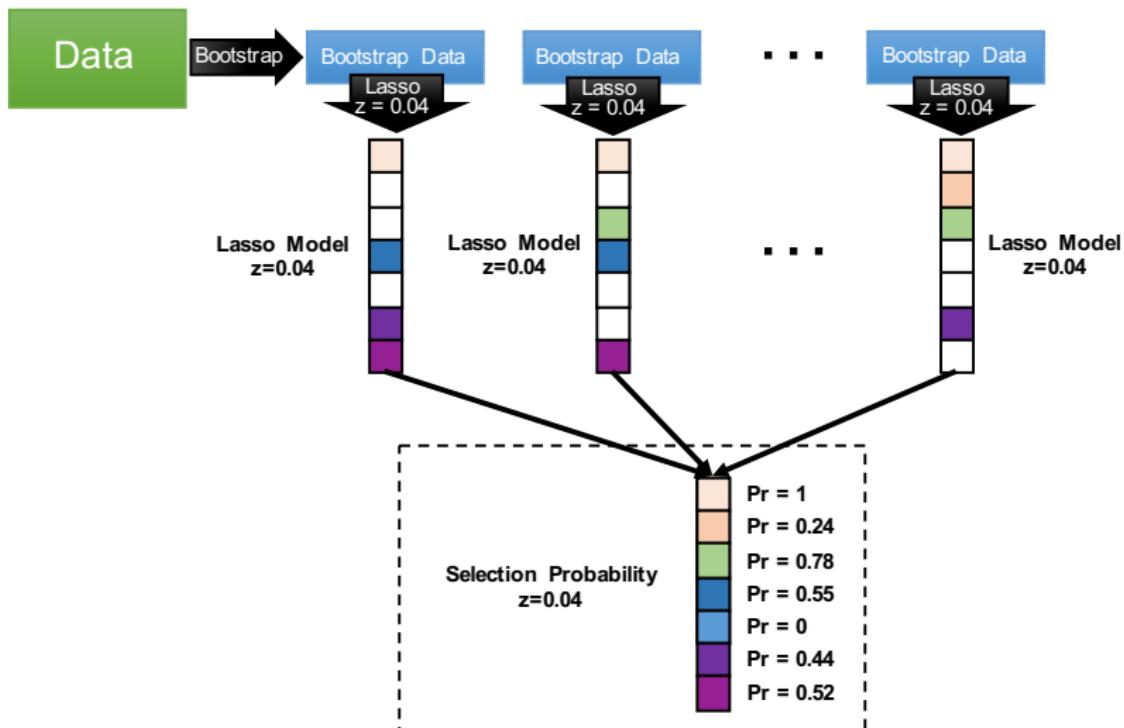
- Network Construction
- Matrix Completion
- Multi-Task Learning
- Sparse inverse covariance
- Complicated feature structures (graph, tree, etc.)
- ...

# Distill Sparsity?

- Sparse learning includes a set of learning algorithms with embedded feature selection.
- What if we would like to use only the features picked by sparse learning algorithms?
  - Methods such as Lasso may be sensitive to data perturbation (especially correlated design)
  - If we can rank features in a stable manner, then we are able to use such rank in a filter/wrapper fashion.

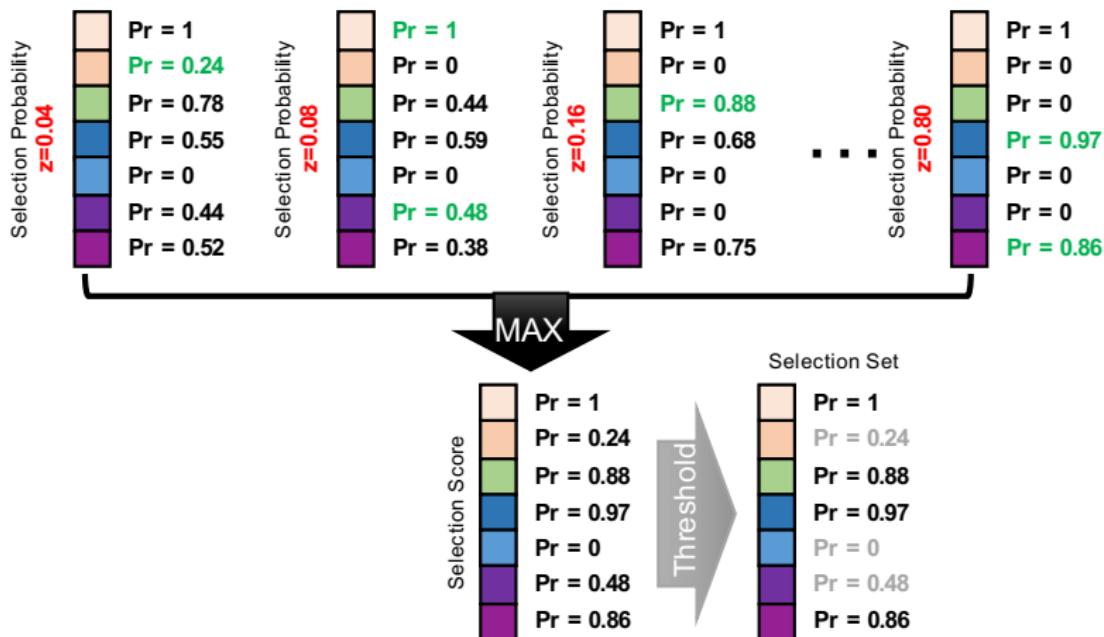
# Stability Selection

## Step 1: Computing Selection Probability



# Stability Selection

## Step 2: Thresholding the Selection Score



# Stability Selection

Stability selection is a principled procedure with proven properties on how many irrelevant features will be selected in this way.

**Theorem 1 (Error control)** Assume that the distribution of  $\{1_{\{k \in \hat{S}^\lambda\}}, k \in N\}$  is exchangeable for all  $\lambda \in \Lambda$ . Also, assume that the original procedure is not worse than random guessing, i.e. for any  $\lambda \in \Lambda$ ,

$$\frac{E(|S \cap \hat{S}^\lambda|)}{E(|N \cap \hat{S}^\lambda|)} \geq \frac{|S|}{|N|}. \quad (8)$$

The expected number  $V$  of falsely selected variables is then bounded by

$$E(V) \leq \frac{1}{2\pi_{thr} - 1} \frac{q_\Lambda^2}{p}. \quad (9)$$

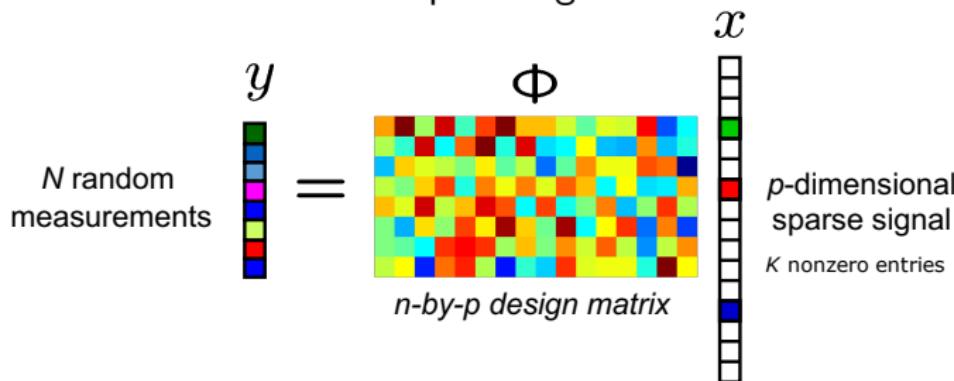
Meinshausen, Nicolai, and Peter Bühlmann. “Stability selection.” Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72.4 (2010): 417-473.

# Algorithms

# Compressive Sensing Revisit

## Problem Setting

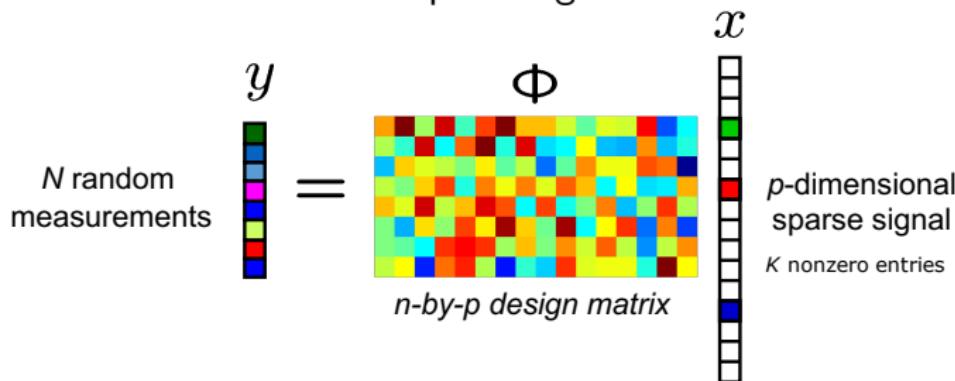
- When signal is sparse/compressible, we acquire a **condensed representation**  $y = \Phi x$
- We would like to recover the sparse signal  $x$ .



# Compressive Sensing Revisit

## Problem Setting

- When signal is sparse/compressible, we acquire a **condensed representation**  $y = \Phi x$
- We would like to recover the sparse signal  $x$ .



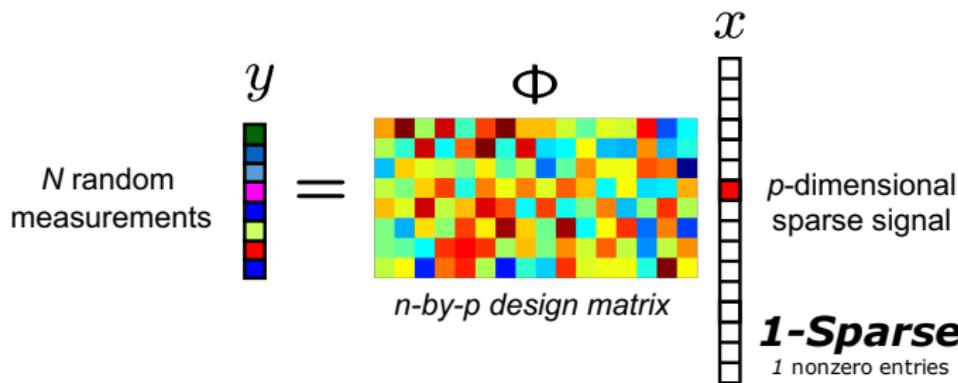
- Signal recovery via cardinality  $\ell_0$  optimization  

$$x^* = \arg \min_{y=\Phi x} \|x\|_0$$
 or its relaxed version:

$$\min_x \|y - \Phi x\|^2, \quad \text{s.t. } \|x\|_0 \leq N$$

# Greedy Recovery Algorithm (1-Sparse)

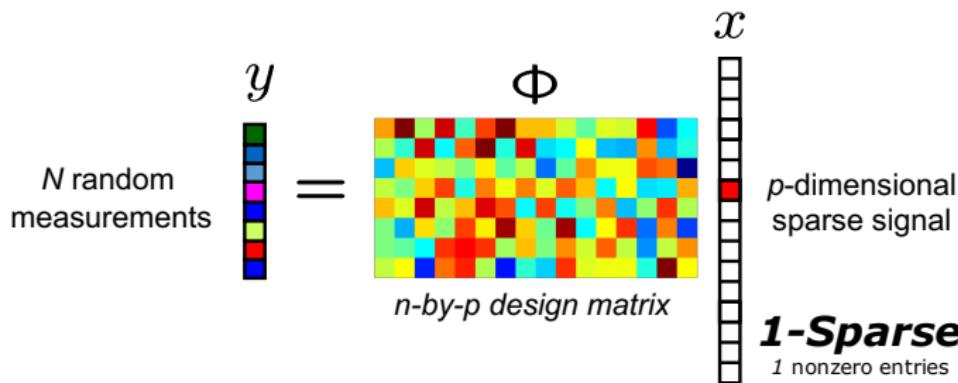
- Consider the following problem



- Can we recover the **support** at 1-Sparse (only one support)?

# Greedy Recovery Algorithm (1-Sparse)

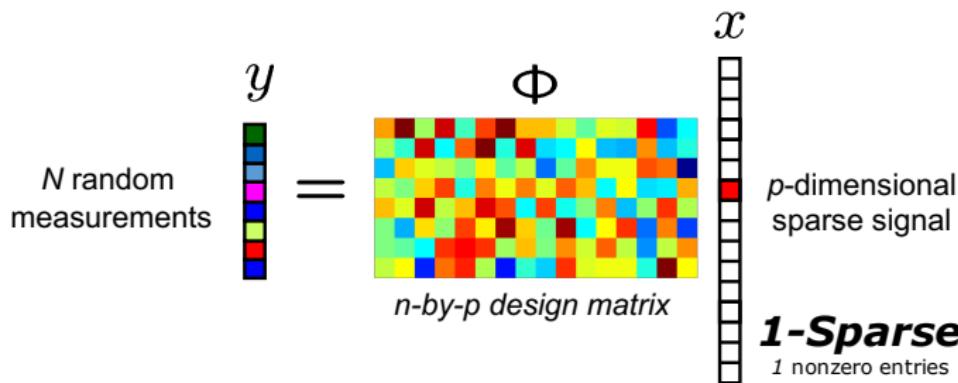
- Consider the following problem



- Can we recover the **support** at 1-Sparse (only one support)?
- If  $\Phi = [\phi_1, \phi_2, \dots, \phi_p]$ , then  $\arg \max |\phi_i^T y|$  gives the support of  $x$

# Greedy Recovery Algorithm (1-Sparse)

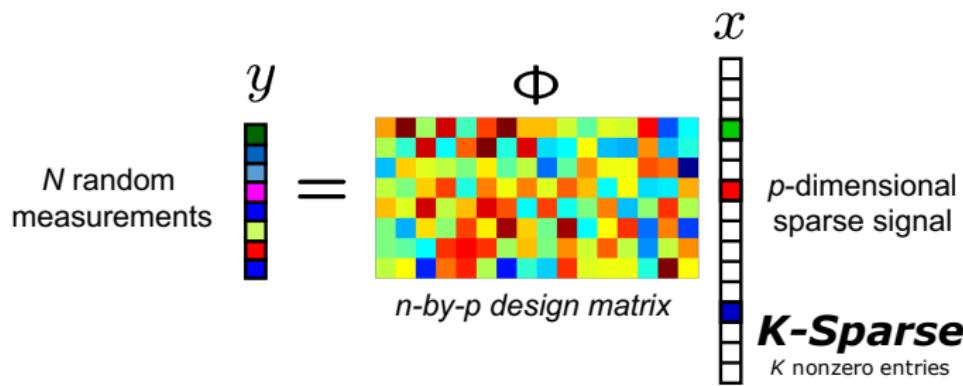
- Consider the following problem



- Can we recover the **support** at 1-Sparse (only one support)?
- If  $\Phi = [\phi_1, \phi_2, \dots, \phi_p]$ , then  $\arg \max |\phi_i^T y|$  gives the support of  $x$
- How to extend to  $K$ -sparse?

# Greedy Recovery Algorithm ( $K$ -Sparse) Matching Pursuit

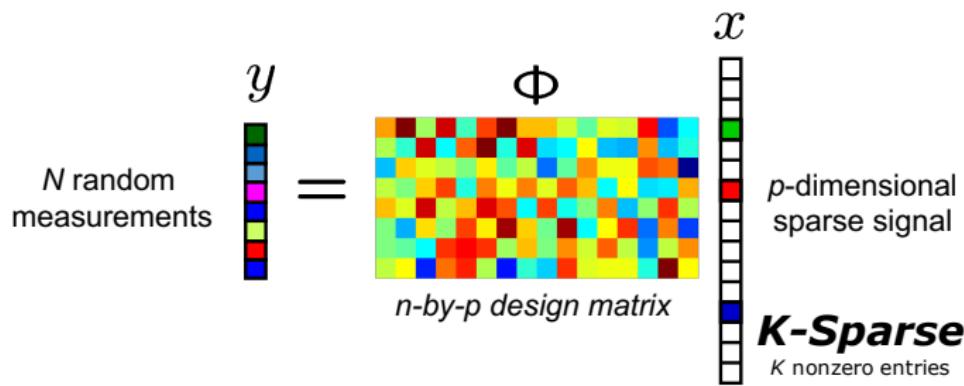
Matching Pursuit [Mallat and Zhang, 1993]



- Find atom  $k = \arg \max_i |\phi_i^T r|$

# Greedy Recovery Algorithm ( $K$ -Sparse) Matching Pursuit

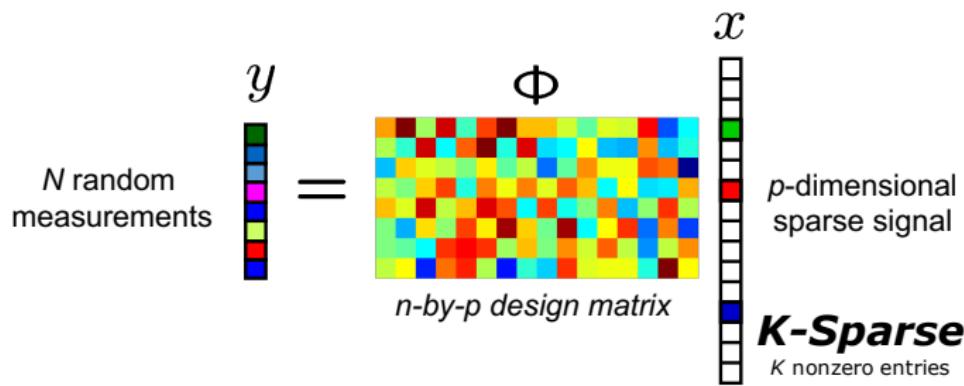
Matching Pursuit [Mallat and Zhang, 1993]



- Find atom  $k = \arg \max_i |\phi_i^T r|$
- Coefficient  $a_k = \phi_k^T r$

# Greedy Recovery Algorithm ( $K$ -Sparse) Matching Pursuit

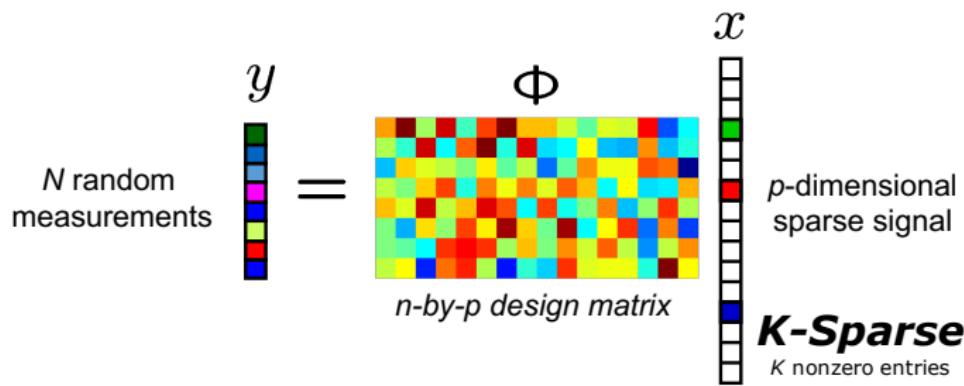
Matching Pursuit [Mallat and Zhang, 1993]



- Find atom  $k = \arg \max_i |\phi_i^T r|$
- Coefficient  $a_k = \phi_k^T r$
- Residue  $r = r - a_k \phi_k$

# Greedy Recovery Algorithm ( $K$ -Sparse) Matching Pursuit

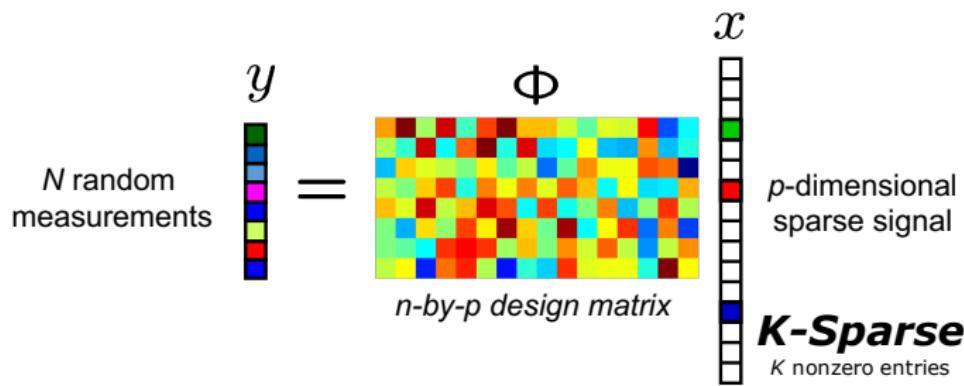
Matching Pursuit [Mallat and Zhang, 1993]



- Find atom  $k = \arg \max_i |\phi_i^T r|$
- Coefficient  $a_k = \phi_k^T r$
- Residue  $r = r - a_k \phi_k$
- Until stop condition (e.g.  $\|r\| < \text{threshold}$ )

# Greedy Recovery Algorithm ( $K$ -Sparse) Matching Pursuit

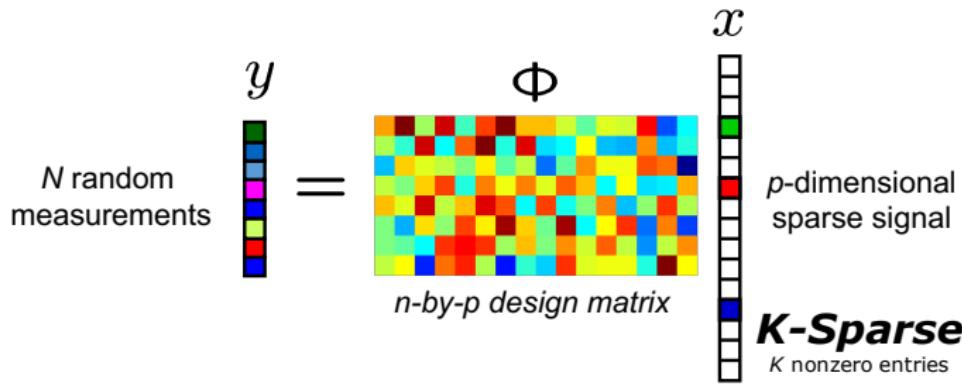
Matching Pursuit [Mallat and Zhang, 1993]



- Find atom  $k = \arg \max_i |\phi_i^T r|$
- Coefficient  $a_k = \phi_k^T r$
- Residue  $r = r - a_k \phi_k$
- Until stop condition (e.g.  $\|r\| < \text{threshold}$ )
- Matching Pursuit

# Greedy Recovery Algorithm ( $K$ -Sparse)

Orthogonal Matching Pursuit [Tropp and Gilbert 2007]

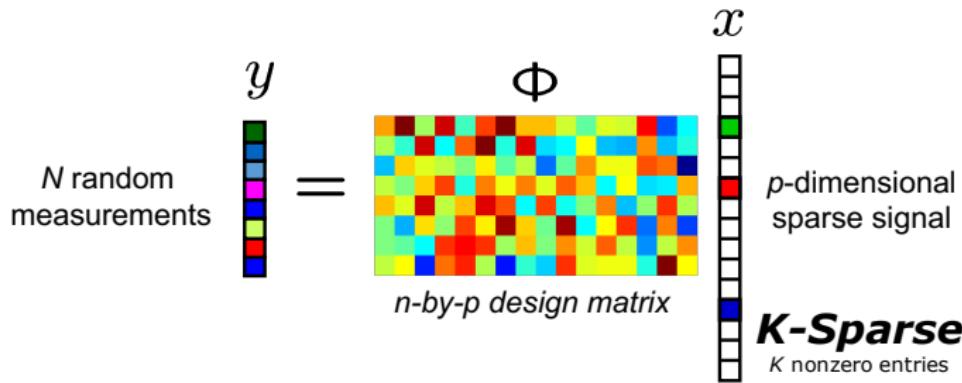


Initialize index set  $S = \{\}$ :

- Residue  $r = y - \Phi_S x_{k-1}$  (Orthogonal projection)

# Greedy Recovery Algorithm ( $K$ -Sparse)

Orthogonal Matching Pursuit [Tropp and Gilbert 2007]

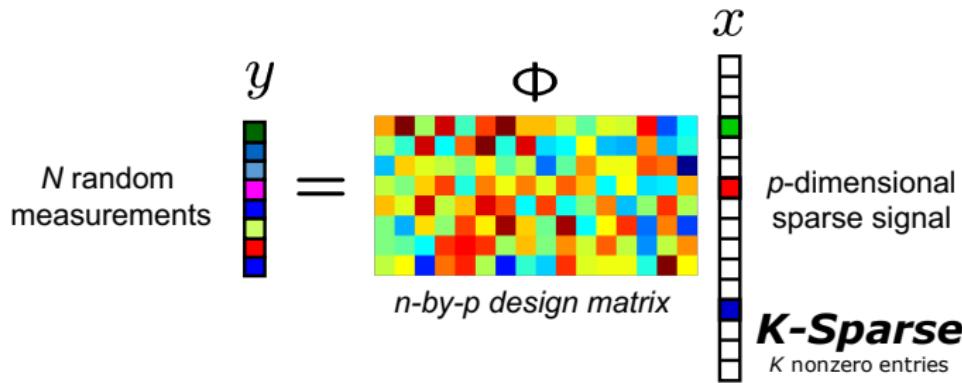


Initialize index set  $S = \{\}$ :

- Residue  $r = y - \Phi_S x_{k-1}$  (Orthogonal projection)
- Find atom  $k = \arg \max_i |\phi_i^T r|$

# Greedy Recovery Algorithm ( $K$ -Sparse)

Orthogonal Matching Pursuit [Tropp and Gilbert 2007]

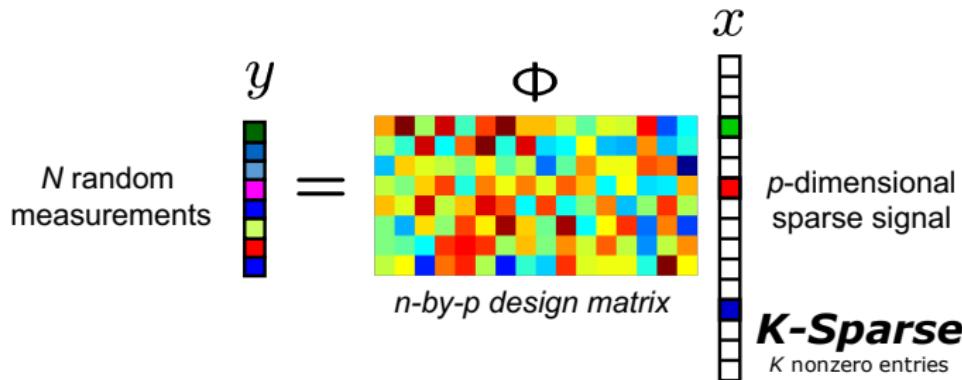


Initialize index set  $S = \{\}$ :

- Residue  $r = y - \Phi_S x_{k-1}$  (Orthogonal projection)
- Find atom  $k = \arg \max_i |\phi_i^T r|$
- Add atom to support  $S = S \cup \{k\}$

# Greedy Recovery Algorithm ( $K$ -Sparse)

Orthogonal Matching Pursuit [Tropp and Gilbert 2007]

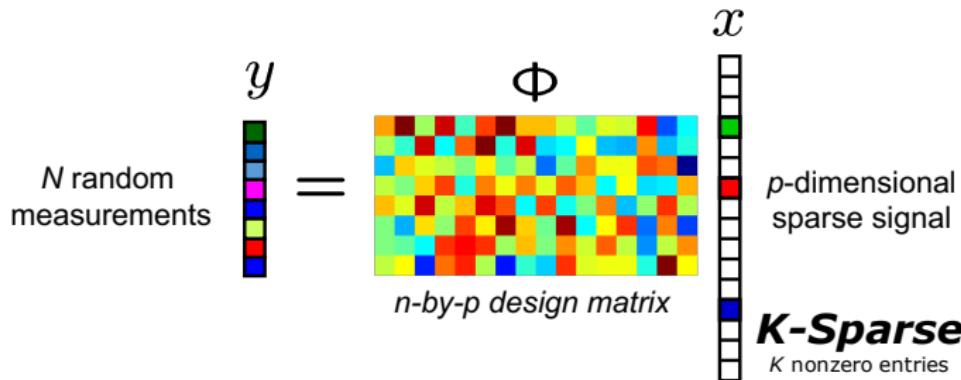


Initialize index set  $S = \{\}$ :

- Residue  $r = y - \Phi_S x_{k-1}$  (Orthogonal projection)
- Find atom  $k = \arg \max_i |\phi_i^T r|$
- Add atom to support  $S = S \cup \{k\}$
- Signal estimate  $x_k = (\Phi_S)^\dagger y$

# Greedy Recovery Algorithm ( $K$ -Sparse)

Orthogonal Matching Pursuit [Tropp and Gilbert 2007]



Initialize index set  $S = \{\}$ :

- Residue  $r = y - \Phi_S x_{k-1}$  (Orthogonal projection)
- Find atom  $k = \arg \max_i |\phi_i^T r|$
- Add atom to support  $S = S \cup \{k\}$
- Signal estimate  $x_k = (\Phi_S)^\dagger y$

orthogonal projection:  $r = (I - \Phi_S(\Phi_S)^\dagger)y = (I - \Phi_S(\Phi_S^T \Phi_S)^{-1} \Phi_S^T)y$

# Orthogonal Matching Pursuit

Orthogonal Matching Pursuit [Tropp and Gilbert 2007]

- Better results than standard matching pursuit
- More computation required.
- MP and OMP approximate the cardinality ( $\ell_0$ ) constrained optimization problem:

$$\min_x \|y - \Phi x\|^2, \quad \text{s.t. } \|x\|_0 \leq N$$

# Orthogonal Matching Pursuit

Orthogonal Matching Pursuit [Tropp and Gilbert 2007]

- Better results than standard matching pursuit
- More computation required.
- MP and OMP approximate the cardinality ( $\ell_0$ ) constrained optimization problem:

$$\min_x \|y - \Phi x\|^2, \quad \text{s.t. } \|x\|_0 \leq N$$

- Signal recovery via  $\ell_1$  optimization  $x^* = \arg \min_{y=\Phi x} \|x\|_1$   
[Candes, Romberg, Tao; Donoho]

# Unified Optimization Framework

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) = \mathcal{L}(x) + \mathcal{R}(x) = \frac{1}{n} \sum_{i=1}^n \ell_i(x) + \mathcal{R}(x) \right\}$$

Name	Loss function $\ell_i(x)$
Least squares	$\frac{1}{2}(y_i - a_i^T x)^2$
Logistic regression	$\log(1 + \exp(-y_i a_i^T x))$
Squared Hinge Loss	$\max(0, 1 - y_i a_i^T x)^2$
Name	Regularizer (penalty) $\mathcal{R}(x)$
Lasso	$\lambda \sum_{j=1}^d  x_j $
Fused Lasso	$\lambda_1 \sum_{j=1}^d  x_j  + \lambda_2 \sum_{j=1}^{d-1}  x_j - x_{j+1} $
Graph Fused Lasso	$\lambda_1 \sum_{j=1}^d  x_j  + \lambda_2 \sum_{(j,k) \in \mathcal{G}}  x_j - x_k $
Group Lasso	$\lambda \sum_{k=1}^K \ x_{\mathcal{G}_k}\ $
Sparse Group Lasso	$\lambda_1 \sum_{j=1}^d  x_j  + \lambda_2 \sum_{k=1}^K \ x_{\mathcal{G}_k}\ $
Tree Lasso	$\sum_{j=1}^J \sum_{k=1}^{K_j} \lambda_k^j \ x_{\mathcal{G}_k^j}\ $

# Gradient Descent for the Composite Model

(Nesterov, 2007; Beck and Teboulle, 2009)

- Optimization objective

$$\min_x f(x) = \mathcal{L}(x) + \lambda \mathcal{R}(x)$$

# Gradient Descent for the Composite Model

(Nesterov, 2007; Beck and Teboulle, 2009)

- Optimization objective

$$\min_x f(x) = \mathcal{L}(x) + \lambda \mathcal{R}(x)$$

- At each iteration we construct a model

$$\mathcal{M}(x_i, \gamma_i) = [\mathcal{L}(x_i) + \nabla \mathcal{L}(x_i)^T (x - x_i)] + \frac{1}{2\gamma_i} \|x - x_i\|_2^2 + \lambda \mathcal{R}(x)$$

Terms: 1st order Taylor expansion, regularization, non-smooth part.  
 $\gamma_i$  is obtained via a certain line search algorithm.

# Gradient Descent for the Composite Model

(Nesterov, 2007; Beck and Teboulle, 2009)

- Optimization objective

$$\min_x f(x) = \mathcal{L}(x) + \lambda \mathcal{R}(x)$$

- At each iteration we construct a model

$$\mathcal{M}(x_i, \gamma_i) = [\mathcal{L}(x_i) + \nabla \mathcal{L}(x_i)^T (x - x_i)] + \frac{1}{2\gamma_i} \|x - x_i\|_2^2 + \lambda \mathcal{R}(x)$$

Terms: 1st order Taylor expansion, regularization, non-smooth part.

$\gamma_i$  is obtained via a certain line search algorithm.

- Optimization algorithm

- Repeat
- $x_{i+1} = \arg \min \mathcal{M}(x_i, \gamma_i)$
- Until convergence

Convergence rate  $O(1/N)$ .

# First Order Optimization

## Proximal Gradient

$$\begin{aligned} x_{i+1} &= \arg \min_x \mathcal{L}(x_i) + \nabla \mathcal{L}(x_i)^T (x - x_i) + \frac{1}{2\gamma_i} \|x - x_i\|_2^2 + \lambda \mathcal{R}(x) \\ &= \arg \min_x \left\{ \frac{1}{2} \|x - (x_i - \gamma_i \nabla \mathcal{L}(x_i))\|^2 + \gamma_i \lambda \mathcal{R}(x) \right\} \\ &\equiv \text{Prox}_{\gamma_i}^{\lambda \mathcal{R}}(x_i - \gamma_i \nabla \mathcal{L}(x_i)) \end{aligned}$$

# First Order Optimization

## Proximal Gradient

$$\begin{aligned} x_{i+1} &= \arg \min_x \mathcal{L}(x_i) + \nabla \mathcal{L}(x_i)^T (x - x_i) + \frac{1}{2\gamma_i} \|x - x_i\|_2^2 + \lambda \mathcal{R}(x) \\ &= \arg \min_x \left\{ \frac{1}{2} \|x - (x_i - \gamma_i \nabla \mathcal{L}(x_i))\|^2 + \gamma_i \lambda \mathcal{R}(x) \right\} \\ &\equiv \text{Prox}_{\gamma_i}^{\lambda \mathcal{R}}(x_i - \gamma_i \nabla \mathcal{L}(x_i)) \end{aligned}$$

- Can be further accelerated with little extra costs
  - FISTA, SpaRSA
  - Convergence rate  $O(1/N^2)$

# First Order Optimization

## Proximal Gradient

$$\begin{aligned} x_{i+1} &= \arg \min_x \mathcal{L}(x_i) + \nabla \mathcal{L}(x_i)^T (x - x_i) + \frac{1}{2\gamma_i} \|x - x_i\|_2^2 + \lambda \mathcal{R}(x) \\ &= \arg \min_x \left\{ \frac{1}{2} \|x - (x_i - \gamma_i \nabla \mathcal{L}(x_i))\|^2 + \gamma_i \lambda \mathcal{R}(x) \right\} \\ &\equiv \text{Prox}_{\gamma_i}^{\lambda \mathcal{R}}(x_i - \gamma_i \nabla \mathcal{L}(x_i)) \end{aligned}$$

- Can be further accelerated with little extra costs
  - FISTA, SpaRSA
  - Convergence rate  $O(1/N^2)$
- How to efficiently solve the proximal operator problem?
- Closed-form solution for  $\ell_1$ ,  $\ell_1/\ell_2$ , analytical form for trace norm

# First Order Optimization

## Proximal Gradient

$$\begin{aligned}
 x_{i+1} &= \arg \min_x \mathcal{L}(x_i) + \nabla \mathcal{L}(x_i)^T (x - x_i) + \frac{1}{2\gamma_i} \|x - x_i\|_2^2 + \lambda \mathcal{R}(x) \\
 &= \arg \min_x \left\{ \frac{1}{2} \|x - (x_i - \gamma_i \nabla \mathcal{L}(x_i))\|^2 + \gamma_i \lambda \mathcal{R}(x) \right\} \\
 &\equiv \text{Prox}_{\gamma_i}^{\lambda \mathcal{R}}(x_i - \gamma_i \nabla \mathcal{L}(x_i))
 \end{aligned}$$

- Can be further accelerated with little extra costs
  - FISTA, SpaRSA
  - Convergence rate  $O(1/N^2)$
- How to efficiently solve the proximal operator problem?
- Closed-form solution for  $\ell_1$ ,  $\ell_1/\ell_2$ , analytical form for trace norm
- Can be extended to second order proximal optimization
  - Subproblem would require FISTA/SpaRSA

# Stochastic Optimization

## Stochastic Proximal Gradient

- Randomly pick a sample  $j \in \{1, \dots, n\}$
- Evaluate the gradient on the  $j$ -th sample  $\nabla \ell_j(x_i)$  and generate a sequence  $\{x_i\}$  via

$$\begin{aligned} x_{i+1} &= \arg \min_x \mathcal{L}(x_i) + \nabla \ell_j(x_i)^T (x - x_i) + \frac{1}{2\gamma_i} \|x - x_i\|_2^2 + \lambda \mathcal{R}(x) \\ &= \arg \min_x \left\{ \frac{1}{2} \|x - (x_i - \gamma_i \nabla \ell_j(x_i))\|^2 + \gamma_i \lambda \mathcal{R}(x) \right\} \\ &\equiv \text{Prox}_{\gamma_i}^{\lambda \mathcal{R}}(x_i - \gamma_i \nabla \ell_j(x_i)) \end{aligned}$$

- Can be extended to using mini-batch
- The use of an expected gradient may introduce variance in each iteration

# Sparse Learning Package



**SLEP** Sparse Learning with Efficient Projections  
Authors: Jun Liu, Shuiwang Ji, Jieping Ye

[SLEP 4.1 Download](#) | [SLEP 4.1 Manual](#)

- 1** First-Order Method. At each iteration, we only need to evaluate the function value and the gradient; and thus the algorithms can handle large-scale sparse data.
- 2** Optimal Convergence Rate. The convergence rate  $O(1/k^2)$  is optimal for smooth convex optimization via the first-order black-box methods.
- 3** Efficient Projection. The projection problem (proximal operator) can be solved efficiently.
- 4** Pathwise Solutions. The SLEP package provides functions that efficiently compute the pathwise solutions corresponding to a series of regularization parameters by the "warm-start" technique.

Website: <http://www.yelab.net/software/SLEP/>  
Github Version: <https://github.com/jiayuzhou/SLEP/>