MLCC 2015
Variable Selection and Sparsity

Lorenzo Rosasco
UNIGE-MIT-IIT

June 24, 2015

Outline

Variable Selection

Subset Selection

Greedy Methods: (Orthogonal) Matching Pursuit

Convex Relaxation: LASSO & Elastic Net

Prediction and Interpretability

- In many practical situations, beyond prediction, it is important to obtain **interpretable** results

Prediction and Interpretability

- In many practical situations, beyond prediction, it is important to obtain **interpretable** results
- Interpretability is often determined by detecting which factors allow good **prediction**

Prediction and Interpretability

- In many practical situations, beyond prediction, it is important to obtain **interpretable** results
- Interpretability is often determined by detecting which factors allow good **prediction**

We look at this question from the perspective of **variable selection**

Linear Models

Consider a linear model

$$f_w(x) = w^T x = \sum_{i=1}^{v} w^j x^j$$

Here

- the components x^j of an input can be seen as **measurements** (pixel values, dictionary words count, gene expressions, . . .)

Linear Models

Consider a linear model

$$f_w(x) = w^T x = \sum_{i=1}^{v} w^j x^j$$

Here

- the components x^j of an input can be seen as **measurements** (pixel values, dictionary words count, gene expressions, ...)

- Given data, the goal of variable selection is to detect which are **variables important for prediction**

Linear Models

Consider a linear model

$$f_w(x) = w^T x = \sum_{i=1}^{v} w^j x^j$$

Here

- the components x^j of an input can be seen as **measurements** (pixel values, dictionary words count, gene expressions, ...)

- Given data, the goal of variable selection is to detect which are **variables important for prediction**

Key assumption: the best possible prediction rule is **sparse**, that is only few of the coefficients are non zero

Outline

Variable Selection

Subset Selection

Greedy Methods: (Orthogonal) Matching Pursuit

Convex Relaxation: LASSO & Elastic Net

Linear Models

Consider a linear model

$$f_w(x) = w^T x = \sum_{i=1}^{v} w^j x^j \tag{1}$$

Here

- the components $x^j$of an input are specific **measurements** (pixel values, dictionary words count, gene expressions, ...)

Linear Models

Consider a linear model

$$f_w(x) = w^T x = \sum_{i=1}^{v} w^j x^j \tag{1}$$

Here

- the components x^j of an input are specific **measurements** (pixel values, dictionary words count, gene expressions, ...)

- Given data the goal of variable selection is to detect which **variables important for prediction**

Linear Models

Consider a linear model

$$f_w(x) = w^T x = \sum_{i=1}^{v} w^j x^j \tag{1}$$

Here

- the components x^j of an input are specific **measurements** (pixel values, dictionary words count, gene expressions, ...)

- Given data the goal of variable selection is to detect which **variables important for prediction**

Key assumption: the best possible prediction rule is **sparse**, that is only few of the coefficients are non zero

Notation

We need some notation:

- X_n be the n by D data matrix

Notation

We need some notation:

- X_n be the n by D data matrix
- i $X^j \in \mathbb{R}^n$, $j = 1, \dots, D$ its columns

Notation

We need some notation:

- X_n be the n by D data matrix
- i $X^j \in \mathbb{R}^n$, $j = 1, \ldots, D$ its columns
- $Y_n \in \mathbb{R}^n$ the output vector

High-dimensional Statistics

Estimating a linear model corresponds to solving a linear system

$$X_n w = Y_n.$$

- Classically $n \gg D$ **low dimension/overdetermined system**

High-dimensional Statistics

Estimating a linear model corresponds to solving a linear system

$$X_n w = Y_n.$$

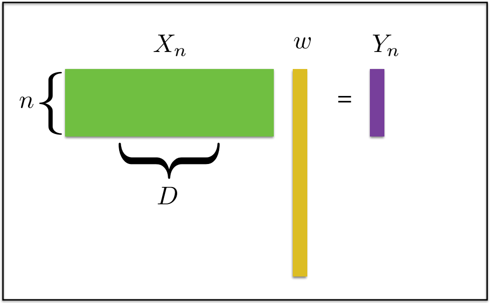
- Classically $n \gg D$ **low dimension/overdetermined system**
- Lately $n \ll D$ **high dimensional/underdetermined system**

Buzzwords: compressed sensing, high-dimensional statistics . . .

High-dimensional Statistics

Estimating a linear model corresponds to solving a linear system

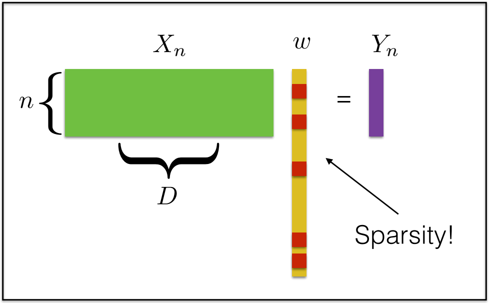
$$X_n w = Y_n.$$



High-dimensional Statistics

Estimating a linear model corresponds to solving a linear system

$$X_n w = Y_n.$$



Brute Force Approach

Sparsity can be measured by the ℓ_0 **norm**

$$\|w\|_0 = |\{j \mid w^j \neq 0\}|$$

that counts non zero components in w

Brute Force Approach

Sparsity can be measured by the ℓ_0 **norm**

$$\|w\|_0 = |\{j \mid w^j \neq 0\}|$$

that counts non zero components in w

If we consider the square loss, it can be shown that a **regularization** approach is given by

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i))^2 + \lambda \|w\|_0$$

The Brute Force Approach is Hard

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i))^2 + \lambda \|w\|_0$$

The above approach is as **hard as a brute force approach**: considering all training sets obtained with all possible subsets of variables (single, couples, triplets... of variables)

The Brute Force Approach is Hard

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i))^2 + \lambda \|w\|_0$$

The above approach is as **hard as a brute force approach**: considering all training sets obtained with all possible subsets of variables (single, couples, triplets... of variables)

The computational complexity is combinatorial. In the following we consider two possible approximate approaches:

- greedy methods
- convex relaxation

Outline

Variable Selection

Subset Selection

Greedy Methods: (Orthogonal) Matching Pursuit

Convex Relaxation: LASSO & Elastic Net

Greedy Methods Approach

Greedy approaches encompasses the following steps:

1. initialize the residual, the coefficient vector, and the index set

Greedy Methods Approach

Greedy approaches encompasses the following steps:

1. initialize the residual, the coefficient vector, and the index set
2. find the variable most correlated with the residual

Greedy Methods Approach

Greedy approaches encompasses the following steps:

1. initialize the residual, the coefficient vector, and the index set
2. find the variable most correlated with the residual
3. update the index set to include the index of such variable

Greedy Methods Approach

Greedy approaches encompasses the following steps:

1. initialize the residual, the coefficient vector, and the index set
2. find the variable most correlated with the residual
3. update the index set to include the index of such variable
4. update/compute coefficient vector

Greedy Methods Approach

Greedy approaches encompasses the following steps:

1. initialize the residual, the coefficient vector, and the index set
2. find the variable most correlated with the residual
3. update the index set to include the index of such variable
4. update/compute coefficient vector
5. update residual.

The simplest such procedure is called forward stage-wise regression in statistics and matching pursuit (MP) in signal processing

Initialization

Let r, w, I denote the residual, the coefficient vector, an index set, respectively.

Initialization

Let r, w, I denote the residual, the coefficient vector, an index set, respectively.

The MP algorithm starts by initializing the residual $r \in \mathbb{R}^n$, the coefficient vector $w \in \mathbb{R}^D$, and the index set $I \subseteq \{1, \dots, D\}$

$$r_0 = Y_n, \quad , w_0 = 0, \quad I_0 = \emptyset$$

Selection

The variable most correlated with the residual is given by

$$k = \arg \max_{j=1,\dots,D} a_j, \quad a_j = \frac{(r_{i-1}^T X^j)^2}{\|X^j\|^2},$$

where we note that

$$v^j = \frac{r_{i-1}^T X^j}{\|X^j\|^2} = \arg\min_{v \in \mathbb{R}} \|r_{i-1} - X^j v\|^2, \quad \|r_{i-1} - X^j v^j\|^2 = \|r_{i-1}\|^2 - a_j$$

Selection (cont.)

Such a selection rule has two interpretations:

- We select the variable with larger **projection** on the output, or equivalently
- we select the variable such that the corresponding column best explains the the output vector in a **least squares sense**

Active Set, Solution and residual Update

Then, index set is updated as $I_i = I_{i-1} \cup \{k\}$, and the coefficients vector is given by

$$w_i = w_{i-1} + w_k, \quad w_k k = v_k e_k$$

where e_k is the element of the canonical basis in $\mathbb{R}^D$with k-th component different from zero

Active Set, Solution and residual Update

Then, index set is updated as $I_i = I_{i-1} \cup \{k\}$, and the coefficients vector is given by

$$w_i = w_{i-1} + w_k, \quad w_k k = v_k e_k$$

where e_k is the element of the canonical basis in $\mathbb{R}^D$with k-th component different from zero

Finally, the residual is updated

$$r_i = r_{i-1} - Xw^k$$

Orthogonal Matching Pursuit

A variant of the above procedure, called Orthogonal Matching Pursuit, is also often considered, where the coefficient computation is replaced by

$$w_i = \arg\min_{w \in \mathbb{R}^D} \|Y_n - X_n M_{I_i} w\|^2,$$

where the D by D matrix M_I is such that $(M_I w)^j = w^j$ if $j \in I$ and $(M_I w)^j = 0$ otherwise. Moreover, the residual update is replaced by

$$r_i = Y_n - X_n w_i$$

Theoretical Guarantees

If

- the solution is sparse, and
- the data matrix has columns "not too correlated"

OMP can be shown to recover with high probability the right vector of coefficients

Outline

Variable Selection

Subset Selection

Greedy Methods: (Orthogonal) Matching Pursuit

Convex Relaxation: LASSO & Elastic Net

ℓ_1 Norm and Regularization

Another popular approach to find sparse solutions is based on a **convex relaxation**
Namely, the ℓ_0 norm is replaced by the ℓ_1 norm,

$$\|w\|_1 = \sum_{j=1}^{D} |w^j|$$

ℓ_1 Norm and Regularization

Another popular approach to find sparse solutions is based on a **convex relaxation**
Namely, the ℓ_0 norm is replaced by the ℓ_1 norm,

$$\|w\|_1 = \sum_{j=1}^{D} |w^j|$$

In the case of least squares, one can consider

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i))^2 + \lambda \|w\|_1$$

Convex Relxation

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i))^2 + \lambda \|w\|_1.$$

- The above problem is called LASSO in statistics and Basis Pursuit in signal processing
- The objective function defining the corresponding minimization problem is convex but not differentiable
- Tools from non-smooth convex optimization are needed to find a solution

Iterative Soft Thresholding

A simple yet powerful procedure to compute a solution is based on the so called **iterative soft thresholding algorithm (ISTA)**:

Iterative Soft Thresholding

A simple yet powerful procedure to compute a solution is based on the so called **iterative soft thresholding algorithm (ISTA)**:

$$w_0 = 0, \quad w_i = S_{\lambda\gamma}(w_{i-1} - \frac{2\gamma}{n} X_n^T (Y_n - X_n w_{i-1})), \quad i = 1, \ldots, T_{\mathsf{max}}$$

Iterative Soft Thresholding

A simple yet powerful procedure to compute a solution is based on the so called **iterative soft thresholding algorithm (ISTA)**:

$$w_0 = 0, \quad w_i = S_{\lambda\gamma}(w_{i-1} - \frac{2\gamma}{n} X_n^T (Y_n - X_n w_{i-1})), \quad i = 1, \ldots, T_{\mathsf{max}}$$

At each iteration a non linear soft thresholding operator is applied to a gradient step

Iterative Soft Thresholding (cont.)

$$w_0 = 0, \quad w_i = S_{\lambda\gamma}(w_{i-1} - \frac{2\gamma}{n} X_n^T (Y_n - X_n w_{i-1})), \quad i = 1, \dots, T_{\text{max}}$$

- the iteration should be run until a convergence criterion is met, e.g. $\|w_i - w_{i-1}\| \leq \epsilon$, for some precision ϵ, or a maximum number of iteration T_{max} is reached
- To ensure convergence we should choose the step-size

$$\gamma = \frac{n}{2\|X_n^T X_n\|}$$

Splitting Methods

In ISTA the contribution of error and regularization are **split**:

- the argument of the soft thresholding operator corresponds to a step of gradient descent

$$\frac{2}{n} X_n^T (Y_n - X_n w_{i-1})$$

- The soft thresholding operator depends only on the regularization and acts component wise on a vector w, so that

$$S_\alpha(u) = ||u| - \alpha|_+ \frac{u}{|u|}.$$

Soft Thresholding and Sparsity

$$S_\alpha(u) = ||u| - \alpha|_+ \frac{u}{|u|}.$$

The above expression shows that the coefficients of the solution computed by ISTA can be exactly zero

Soft Thresholding and Sparsity

$$S_\alpha(u) = ||u| - \alpha|_+ \frac{u}{|u|}.$$

The above expression shows that the coefficients of the solution computed by ISTA can be exactly zero

This can be contrasted to Tikhonov regularization where this is hardly the case

Lasso meets Tikhonov: Elastic Net

Indeed, it is possible to see that:

- while Tikhonov allows to compute a stable solution, in general its solution is not sparse
- On the other hand the solution of LASSO, might not be stable

Lasso meets Tikhonov: Elastic Net

Indeed, it is possible to see that:

- while Tikhonov allows to compute a stable solution, in general its solution is not sparse
- On the other hand the solution of LASSO, might not be stable

The elastic net algorithm, defined as

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i))^2 + \lambda(\alpha \|w\|_1 + (1-\alpha)\|w\|_2^2), \quad \alpha \in [0,1] \quad (2)$$

can be seen as hybrid algorithm which interpolates between Tikhonov and LASSO

ISTA for Elastic Net

The ISTA procedure can be adapted to solve the elastic net problem, where the gradient descent step incorporates also the derivative of the ℓ^2 penalty term. The resulting algorithm is

$$\begin{aligned} w_0 &= 0, \\ \texttt{for} \quad & i = 1, \ldots, T_{\mathsf{max}} \\ w_i &= S_{\lambda\alpha\gamma}((1 - \lambda\gamma(1-\alpha))w_{i-1} - \frac{2\gamma}{n} X_n^T (Y_n - X_n w_{i-1})), \end{aligned}$$

To ensure convergence we should choose the step-size

$$\gamma = \frac{n}{2(\|X_n^T X_n\| + \lambda(1-\alpha))}$$

Wrapping Up

Next Class