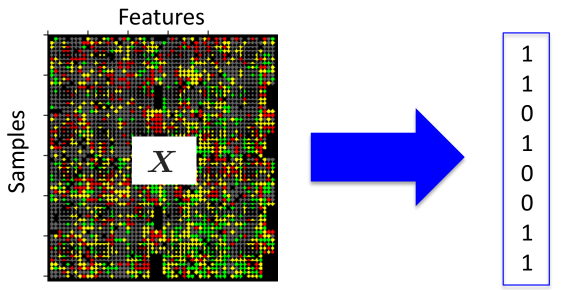
Feature Selection and Sparsity

Makoto Yamada
makoto.yamada@oist.jp
Okinawa Institute of Science and Technology

Introduction

Feature selection is important for high-dimensional data:

- User data ($d > 100$), e.g., e-mail spam detection.
- Gene expression data ($d > 20000$), e.g., cancer classification.
- Text based feature such as TF-IDF ($d > 100,000$)



Motivation1

The purpose of feature selection is

- to improve the prediction accuracy by getting rid of non-important features.
- to make the prediction faster.
- to interpret data.
- to handle high-dimensional data.

Motivation2

Let us think about the least-squared regression problem:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top \in \mathbb{R}^d$,
$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \in \mathbb{R}^{d \times n}$, $\boldsymbol{w} = (w_1, w_2, \ldots, w_d)^\top \in \mathbb{R}^d$,
$\boldsymbol{y} \in \mathbb{R}^n$, and $\| \cdot \|_2^2$ is the ℓ_2 norm.

Question:

- $d < n$ and the rank of $\boldsymbol{X}$ is d. Please derive the analytical solution of $\boldsymbol{w}$.

Motivation2

Take the derivative with respect to $\boldsymbol{w}$ and set it to zero:

$$\frac{\partial}{\partial \boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 = -2\boldsymbol{X}(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}) = \boldsymbol{0}$$

Use Eq. (84) of [1]. The solution is given as

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{X}\boldsymbol{y}.$$

If the rank of $\boldsymbol{X}$ is d, $\boldsymbol{X}\boldsymbol{X}^\top$ is invertible.

What happens if the rank of $\boldsymbol{X}$ is less than d?

- $\boldsymbol{X}\boldsymbol{X}^\top$ is not invertible.

A possible solution is to use feature selection! If we select $r < d$ features, we can compute $\boldsymbol{w}$.

Problem formulation

Problem formulation of feature selection:

- Input vector: $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top \in \mathbb{R}^d$
- Output: $y \in \mathbb{R}$
- Paired data: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$

Goal: Select $r (r < d)$ features of input $\boldsymbol{x}$ that are responsible for output y.

Problems: There is 2^d combinations :(It is hard even if d is 100.

Feature Selection Algorithms

The feature selection algorithms can be categorized into three types.

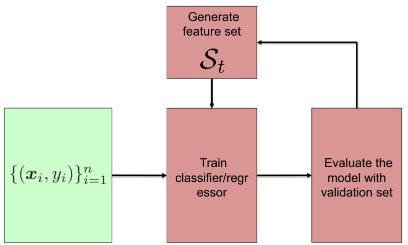
- Wrapper Method
  Use a predictive model to select features.
- Filter Method
  Use a proxy measure (such as mutual information) instead of the error rate to select features.
- Embedded Method
  Features are selected as part of the model construction process.

Wrapper Method

Use a predictive model (e.g., classifier) to select features.

The simplest approach would be...

1. Generate feature set $\mathcal{S}_t$
2. Train predictive model with $\mathcal{S}_t$ and test the prediction accuracy with hold-out set.
3. Iterate 1 and 2 until all feature combination is examined.



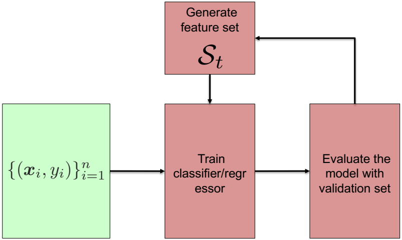
Wrapper Method

Pro:

- It can select features that have feature-feature interaction.

Cons:

- Computationally expensive (2^d combination).



Filter Method

Use a proxy measure (such as mutual information) instead of the error rate to select features.

Pros:

- It scales well.
- Can select features from high-dimensional data (both linear and nonlinear way).

Cons:

- The feature selection is independent of the model. The selected features may not be the best set to achieve highest accuracy.
- It is hard to detect select features with interaction.

Filter Method (Example)

Maximum Relevance Feature Selection (MR)

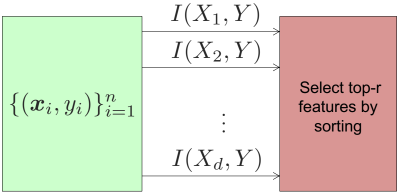
Compute association score between each feature and its output and rank them.

- Correlation, Mutual information, and the kernel based independence measures are used.
- Easy to implement and it scales well.

Optimization problem:

$$\max_{\boldsymbol{\beta} \in \{0,1\}^d} \frac{1}{S} \sum_{k=1}^{d} \beta_k I(X_k, Y),$$

where $S = \beta_1 + \ldots + \beta_d$.



Filter Method (Example)

Minimum Redundancy Maximum Relevance (mRMR) [2]

MR feature selection tends to select redundant features.

mRMR method is to

- select features that have high association to its output.
- select independent features.

Optimization problem:

$$\max_{\boldsymbol{\beta} \in \{0,1\}^d} \quad \frac{1}{S} \sum_{k=1}^{d} \beta_k I(X_k, Y) - \frac{1}{S^2} \sum_{k=1}^{d} \sum_{k'=1}^{d} \beta_k \beta_{k'} I(X_k, X_{k'}).$$

This optimization problem can be solved by using greedy algorithm.

Filter Method (Mutual Information)

To optimize mRMR, we tend to use the mutual information as an association score.

Independence:

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$$

Mutual Information:

$$\mathsf{MI}(X, Y) = \iint p(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} \mathsf{d}\boldsymbol{x}\mathsf{d}\boldsymbol{y}$$

Under independence:

$$\mathsf{MI}(X, Y) = \iint p(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p(\boldsymbol{x})p(\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} \mathsf{d}\boldsymbol{x}\mathsf{d}\boldsymbol{y} = 0$$

Filter Method (Linear Correlation)

To optimize mRMR, we may be able to use the Pearson's correlation coefficient

Pearson's correlation coefficient:

$$\text{PCC}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y},$$

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$$

where $\mu_X = \mathbb{E}[X]$, $\mu_Y = \mathbb{E}[Y]$, $\sigma_X^2 = \mathbb{E}[(X - \mu_X)^2]$, and $\sigma_Y^2 = \mathbb{E}[(Y - \mu_Y)^2]$.

The cross-covariance can be written as

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y].$$

That is, if $\text{PCC}(X, Y) = 0$, $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$

The relationship between independence and correlation

If X and Y are independent, we can write

$$\begin{aligned}\mathbb{E}[XY] &= \iint xy\ p(x,y)\mathrm{d}x\mathrm{d}y, \\ &= \iint xy\ p(x)p(y)\mathrm{d}x\mathrm{d}y, (\textit{independence}) \\ &= \left(\int x\ p(x)\mathrm{d}x\right)\left(\int y\ p(y)\mathrm{d}y\right) \\ &= \mathbb{E}[X]\mathbb{E}[Y]\end{aligned}$$

That is, if X and Y are independent, $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$.
Note that, even if $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$, X and Y can be dependent.

Empirical estimation of Cross-covariance

To optimize mRMR, we may be able to use the Pearson's correlation coefficient

Cross-Covariance (population):

$$\mathsf{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$$

Cross-Covariance estimation:

$$\widehat{\mathsf{Cov}}(X, Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{\mu}_X)(y_i - \widehat{\mu}_Y)$$

$$\widehat{\mu}_X = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\boldsymbol{x}^\top \mathbf{1}_n, \quad \widehat{\mu}_Y = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\boldsymbol{y}^\top \mathbf{1}_n,$$

where $\mathbf{1}_n = (1, 1, \ldots, 1)^\top \in \mathbb{R}^n$ is the vector with all ones.

Empirical estimation of cross-covariance

Cross-Covariance estimation:

$$\widehat{\mathrm{Cov}}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \frac{1}{n}\boldsymbol{x}^\top \mathbf{1}_n)(y_i - \frac{1}{n}\boldsymbol{y}^\top \mathbf{1}_n)$$
$$= \frac{1}{n}\left(\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\boldsymbol{x}^\top \mathbf{1}_n \mathbf{1}_n^\top \boldsymbol{y}\right)$$
$$= \frac{1}{n}\left(\boldsymbol{x}^\top \boldsymbol{y} - \frac{1}{n}\boldsymbol{x}^\top \mathbf{1}_n \mathbf{1}_n^\top \boldsymbol{y}\right)$$
$$= \frac{1}{n}\boldsymbol{x}^\top \left(\boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n \mathbf{1}_n^\top\right)\boldsymbol{y}$$
$$= \frac{1}{n}\boldsymbol{x}^\top \boldsymbol{H}\boldsymbol{y},$$

where $\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n \mathbf{1}_n^\top$ is the centering matrix and $\boldsymbol{I}_n$ is the identity matrix. (Note $\boldsymbol{H}\boldsymbol{H} = \boldsymbol{H}$).

Empirical estimation of covariance

Covariance estimation:

$$
\begin{aligned}
\widehat{\mathrm{Cov}}(X,Y)^2 &= \frac{1}{n^2}\boldsymbol{x}^\top \boldsymbol{H}\boldsymbol{y}\boldsymbol{x}^\top \boldsymbol{H}\boldsymbol{y}, \\
&= \frac{1}{n^2}\mathrm{tr}\left(\boldsymbol{x}^\top \boldsymbol{H}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{H}\boldsymbol{x}\right) \\
&= \frac{1}{n^2}\mathrm{tr}\left(\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{H}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{H}\right) \\
&= \frac{1}{n^2}\mathrm{tr}\left(\boldsymbol{K}\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}\right),
\end{aligned}
$$

where $\boldsymbol{K} = \boldsymbol{x}\boldsymbol{x}^\top \in \mathbb{R}^{n\times n}$ and $\boldsymbol{L} = \boldsymbol{y}\boldsymbol{y}^\top \in \mathbb{R}^{n\times n}$.

Advanced Topic (Hilbert-Schmidt Independence Criterion)

Hilbert Schmidt Independence Criterion (HSIC) [3]

Empirical V-statistics of HSIC is given as

$$\mathrm{HSIC}(X, Y) = \frac{1}{n^2}\mathrm{tr}(\boldsymbol{K}\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}),$$

where we use the Gaussian kernel:

$$\boldsymbol{K}_{ij} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2}\right), \quad \boldsymbol{L}_{ij} = \exp\left(-\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2}{2\sigma^2}\right).$$

HSIC takes 0 if and only if X and Y are independent.

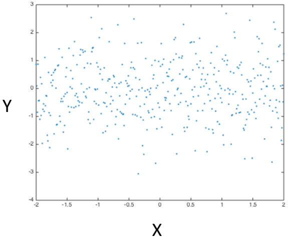
Since we can decompose $\boldsymbol{K} = \boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ and $\boldsymbol{L} = \boldsymbol{\Psi}^\top\boldsymbol{\Psi}$, we have

$$\mathrm{HSIC}(X, Y) = \frac{1}{n^2}\mathrm{tr}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{H}\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\boldsymbol{H}) = \frac{1}{n^2}\|\mathrm{vec}(\boldsymbol{\Psi}\boldsymbol{H}\boldsymbol{\Phi}^\top)\|_2^2 \geq 0$$

Advanced Topic (HSIC)

Hilbert-Schmidt Independence Criterion (HSIC) experiments

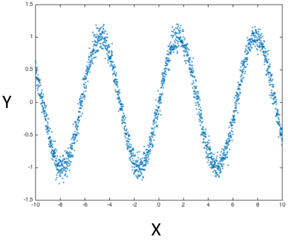
X and Y are independent



NHSIC = 0.0031

Pearson CC= 0.0343

X and Y are dependent



NHSIC = 0.2842

Pearson CC = 0.1983

Embedded Method

Features are selected as part of the model construction process. Embedded method can be regarded as an intermediate method between wrapper and filter methods.

Pros:

- Can select features with high prediction accuracy.
- Computationally efficient than wrapper method.

Cons:

- Computationally expensive than filter method.
- If the input output relationship are nonlinear, it is computationally expensive. It is more suited for linear method.

Embedded Method (Lasso)

Least Absolute Shrinkage and Selection Operator (Lasso)

The optimization problem of Lasso can be written as

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1,$$

where $\|\boldsymbol{w}\|_1 = \sum_{k=1}^{d} |w_k|$ is an ℓ_1 norm.

Lasso is a convex method: The first term is a convex function w.r.t. $\boldsymbol{w}$. ℓ_1 norm (all norm) is convex:

$$\begin{aligned} \|\alpha\boldsymbol{w} + (1-\alpha)\boldsymbol{v}\|_1 &\leq \|\alpha\boldsymbol{w}\|_1 + \|(1-\alpha)\boldsymbol{v}\|_1 \\ &= \alpha\|\boldsymbol{w}\|_1 + (1-\alpha)\|\boldsymbol{v}\|_1 \end{aligned}$$

where $0 \leq \alpha \leq 1$. The sum of two convex functions is convex.

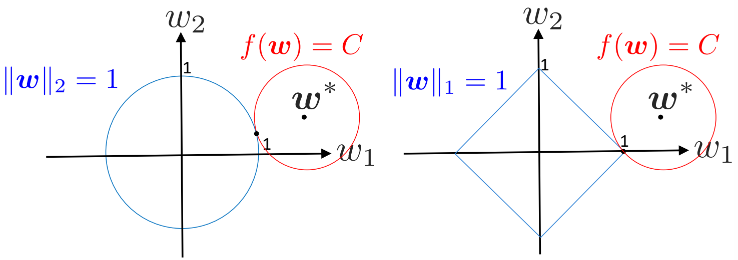
Embedded Method (Lasso)

The ℓ_1 regularization is equivalent to ℓ_1 norm constraint:

$$\min_{\boldsymbol{w}} \quad f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1 \longrightarrow \min_{\boldsymbol{w}} \quad f(\boldsymbol{w}), \quad \text{s.t.} \quad \|\boldsymbol{w}\|_1 \leq \eta.$$

There exists the same solution of the ℓ_1 norm constraint with an arbitrary λ.

Using the ℓ_1 regularizer, we can make $\boldsymbol{w}$ sparse.



L2 regularization (Ridge) L1 regularization (Lasso)

When Lasso helpful?

Let us think about a least-squared regression problems:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2.$$

Take the objective function with respect to $\boldsymbol{w}$ and set it to zero:

$$\frac{\partial}{\partial \boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 = -2\boldsymbol{X}(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}) = \boldsymbol{0}$$

Use Eq. (84) of [1]. The solution is given as

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{X}\boldsymbol{y}.$$

If the rank of $\boldsymbol{X}$ is d, the rank of $\boldsymbol{X}\boldsymbol{X}^\top$ is also d and it is invertible.

What happens if the rank of $\boldsymbol{X}$ is less than d?

Lasso with ADMM (1/8)

Lasso has no closed form solution. Thus, we need to iteratively optimize the problem.

Here, we introduce the Alternating Direction Method of Multipliers (ADMM) [5].

We can rewrite the Lasso optimization problem as

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{z}} \quad & \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{w} = \boldsymbol{z} \end{aligned}$$

The key idea here is to split the main objective and the non-differentiable regularization term. Since the last term $\frac{\rho}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2$ is zero if the constraint is satisfied, this problem is equivalent to the original Lasso problem.

Lasso with ADMM (2/8)

Let us denote the Lagrange multipliers as $\boldsymbol{\gamma} \in \mathbb{R}^d$, we can write a Lagrangian function (called Augmented Lagrangian function) as follows:

$$J(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\gamma}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 + \boldsymbol{\gamma}^\top (\boldsymbol{w} - \boldsymbol{z}) \\ + \lambda \|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2,$$

where $\rho > 0$ is a tuning parameter.

Lasso with ADMM (3/8)

In ADMM, we consider the following optimization problem:

$$\min_{\boldsymbol{w},\boldsymbol{z}} \max_{\boldsymbol{\gamma}} \quad J(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\gamma}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 + \boldsymbol{\gamma}^\top(\boldsymbol{w} - \boldsymbol{z}) + \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2,$$

Since we have the relationship,

$$\max_{\boldsymbol{\gamma}} J(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\gamma}) = \begin{cases} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 & (\boldsymbol{w} = \boldsymbol{z}) \\ \infty & (\text{Otherwise}) \end{cases}$$

The optimization problem is equivalent to the original Lasso problem.

Lasso with ADMM (4/8)

Minimizing $J(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\gamma})$ w.r.t. $\boldsymbol{w}$. If we fix $\boldsymbol{z}$ and $\boldsymbol{\gamma}$ as $\boldsymbol{z}^{(t)}$ and $\boldsymbol{\gamma}^{(t)}$, $J(\boldsymbol{w}, \boldsymbol{z}^{(t)}, \boldsymbol{\gamma}^{(t)})$ is convex w.r.t. $\boldsymbol{w}$. That is,

$$\frac{\partial J(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\gamma})}{\partial \boldsymbol{w}} = -\boldsymbol{X}(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}) + \boldsymbol{\gamma} + \rho(\boldsymbol{w} - \boldsymbol{z}) = \boldsymbol{0}.$$

Here, we can use the following equation (see [1] Eq. (84)):

$$\frac{\partial \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2}{\partial \boldsymbol{w}} = -2\boldsymbol{X}(\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}).$$

Solving it for $\boldsymbol{w}$:

$$\begin{aligned} (\boldsymbol{X}\boldsymbol{X}^\top + \rho \boldsymbol{I})\boldsymbol{w} &= \boldsymbol{X}\boldsymbol{y} - \boldsymbol{\gamma}^{(t)} + \rho \boldsymbol{z}^{(t)} \\ \boldsymbol{w}^{(t+1)} &= (\boldsymbol{X}\boldsymbol{X}^\top + \rho \boldsymbol{I})^{-1}(\boldsymbol{X}\boldsymbol{y} - \boldsymbol{\gamma}^{(t)} + \rho \boldsymbol{z}^{(t)}). \end{aligned}$$

Lasso with ADMM (5/8)

Minimizing $J(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\gamma})$ w.r.t. $\boldsymbol{z}$. If we fix $\boldsymbol{w}$ and $\boldsymbol{\gamma}$ as $\boldsymbol{w}^{(t)}$ and $\boldsymbol{\gamma}^{(t)}$, $J(\boldsymbol{w}^{(t)}, \boldsymbol{z}, \boldsymbol{\gamma}^{(t)})$ is convex w.r.t. $\boldsymbol{z}$.

$$J(\boldsymbol{w}^{(t)}, \boldsymbol{z}, \boldsymbol{\gamma}^{(t)}) = \frac{\rho}{2}\|\boldsymbol{z} - \boldsymbol{w}^{(t)}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 - \boldsymbol{\gamma}^\top \boldsymbol{z} + \text{Const.}$$

$\|\boldsymbol{z}\|_1$ is not differentiable at 0. However, we can analytically solve the problem! Moreover, since there is no interaction in the elements of $\boldsymbol{z}$, we can solve it for each element.

$$J(\boldsymbol{w}^{(t)}, (z_1, \ldots, z_\ell, \ldots, z_d), \boldsymbol{\gamma}^{(t)}) = \frac{\rho}{2}(z_\ell - w_\ell^{(t)})^2 + \lambda|z_\ell| - \gamma_\ell z_\ell + \text{Const.}$$

Lasso with ADMM (6/8)

Case1:

$z_\ell > 0, \rho(z_\ell - w_\ell^{(t)}) + \lambda - \gamma_\ell = 0 \longrightarrow z_\ell = w_\ell^{(t)} + \frac{1}{\rho}(\gamma_\ell - \lambda)$

That is, $z_\ell > 0$ if $w_\ell^{(t)} + \frac{1}{\rho}\gamma_\ell > \frac{\lambda}{\rho}$

Case2:

$z_\ell < 0, \rho(z_\ell - w_\ell^{(t)}) - \lambda - \gamma_\ell = 0 \longrightarrow z_\ell = w_\ell^{(t)} + \frac{1}{\rho}(\gamma_\ell + \lambda)$

That is, $z_\ell < 0$ if $w_\ell^{(t)} + \frac{1}{\rho}\gamma_\ell < -\frac{\lambda}{\rho}$

Case3: $z_\ell = 0$, $0 \in \rho(z_\ell - w_\ell^{(t)}) + \lambda[-1\ 1] - \gamma_\ell \longrightarrow$
$w_\ell + \frac{1}{\rho}\gamma_\ell \in [-\frac{\lambda}{\rho}, \frac{\lambda}{\rho}], (z_\ell = 0)$.

Lasso with ADMM (7/8)

Let us introduce the Soft-Thresholding function:

$$S_\lambda(x) = \begin{cases} x - \lambda & (x > \lambda) \\ 0 & (x \in [-\lambda, \lambda]) \\ x + \lambda & (x < -\lambda) \end{cases},$$

$$= \mathrm{sign}(x) \max(0, |x| - \lambda)$$

Therefore, the update of z_ℓ can be simply written by the soft-thresholding function as

$$\widehat{z}_\ell^{(t+1)} = S_{\frac{\lambda}{\rho}}\left(w_\ell^{(t)} + \frac{1}{\rho}\gamma_\ell\right).$$

Lasso with ADMM (8/8)

Maximizing $J(\boldsymbol{w}, \boldsymbol{z}, \gamma)$ w.r.t. γ. That is the optimization problem can be written as

$$\max_{\gamma} J(\boldsymbol{w}, \boldsymbol{z}, \gamma) = \boldsymbol{\gamma}^\top (\boldsymbol{w} - \boldsymbol{z}).$$

To optimize this problem, since we cannot get the analytical solution, we use the gradient ascent algorithm:

$$\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t)} + \rho(\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}).$$

Thus, the ADMM algorithm for Lasso can be summarized as

$$\begin{aligned}
\boldsymbol{w}^{(t+1)} &= (\boldsymbol{X}\boldsymbol{X}^\top + \rho \boldsymbol{I})^{-1}(\boldsymbol{X}\boldsymbol{y} - \boldsymbol{\gamma}^{(t)} + \rho \boldsymbol{z}^{(t)}) \\
\boldsymbol{z}_\ell^{(t+1)} &= S_{\frac{\lambda}{\rho}}(\boldsymbol{w}^{(t+1)} + \frac{1}{\rho}\gamma) \\
\boldsymbol{\gamma}^{(t+1)} &= \boldsymbol{\gamma}^{(t+1)} + \rho(\boldsymbol{w}^{(t+1)} - \boldsymbol{z}^{(t+1)}).
\end{aligned}$$

Elastic-Net

For Lasso, the number of non-zero features should be smaller than n. How to select $r > n$ variables?

Ans: Use the elastic net regularization [6]:

$$\min_{\boldsymbol{w}} \quad \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 + \lambda(\alpha\|\boldsymbol{w}\|_1 + (1-\alpha)\|\boldsymbol{w}\|_2^2),$$

where $0 \le \alpha \le 1$ and $\lambda > 0$ is a regularization parameter.

$\|\boldsymbol{w}\|_2^2$ is differentiable; we can similarly solve it with ADMM.

$$\boldsymbol{w}^{(t+1)} = (\boldsymbol{X}\boldsymbol{X}^\top + 2\lambda(1-\alpha)\boldsymbol{I} + \rho\boldsymbol{I})^{-1}(\boldsymbol{X}\boldsymbol{y} - \boldsymbol{\gamma}^{(t)} + \rho\boldsymbol{z}^{(t)})$$

$$\boldsymbol{z}_\ell^{(t+1)} = S_{\frac{\lambda\alpha}{\rho}}(\boldsymbol{w}^{(t+1)} + \frac{1}{\rho}\gamma)$$

$$\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t+1)} + \rho(\boldsymbol{w}^{(t+1)} - \boldsymbol{z}^{(t+1)}).$$

Thanks to the ℓ_2 regularization, $\boldsymbol{w}$ tends to be dense.

Summary

- Feature selection: Wrapper method, Filter method, and Embedded method
- Wrapper method (Selecting features that maximize prediction accuracy. Computationally expensive.)
- Filter method (Use mutual information to select features, e.g., MR, mRMR, etc.)
- Embedded method (Selecting features during training. e.g., Lasso)
- Alternating Direction Method of Multipliers (ADMM).

[1] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.

[2] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1237, 2005.

[3] A. Gretton, O. Bousquet, Alex. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *ALT*, 2005.

[4] C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *JMLR*, 13:795–828, 2012.

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato,

Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[6] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.