UVA CS 6316: Machine Learning

Lecture 6 Extra: Optimization for Linear Regression Model with Regularizations

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

Extra Recap

- More about LR Model with Regularizations
  - Ridge Regression
  - Lasso Regression
    - Extra: how to perform training
  - Elastic net
    - Extra: how to perform training

Why Invertible In Ridge Regression?

$$\beta^* = \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}$$

(**NOT AN EASY PROOF from SVD angle**), many concepts, SVD, PCA, Eigenvalues, relation to singular

- NOT AN EASY PROOF If through SVD
  - https://www.quora.com/When-is-the-matrix-frac-1-n-X-T-X-+-lambda-I_-d-times-d-invertible

- The determinant of A is equal to the product of its eigenvalues,

$$|A| = \prod_{i=1}^{n} \lambda_i.$$

- The rank of A is equal to the number of non-zero eigenvalues of A.

Why Invertible In Ridge Regression?

symmetric, positive semi-definite *square* Gram matrix $K = A^T A$ — which can be naturally formed even when A is not square. Perhaps the eigenvalues of K might play a comparably important role for general matrices. Since they are not easily related to the eigenvalues of A — which, in the non-square case, don't even exist — we shall endow them with a new name.

Definition 6.27. The *singular values* $\sigma_1, \ldots, \sigma_r$ of an $m \times n$ matrix A are the positive square roots, $\sigma_i = \sqrt{\lambda_i} > 0$, of the nonzero eigenvalues of the associated Gram matrix $K = A^T A$. The corresponding eigenvectors of K are known as the *singular vectors* of A.

Since K is necessarily positive semi-definite, its eigenvalues are always non-negative, $\lambda_i \geq 0$, which justifies the positivity of the singular values of A — independently of whether A itself has positive, negative, or even complex eigenvalues — or is rectangular and has no eigenvalues at all. The standard convention is to label the singular values in decreasing order, so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. Thus, σ_1 will always denote the largest or

Extra: two forms of Ridge Regression

- Totally equivalent

$$\begin{cases} ① \ \underset{\theta}{\text{argmin}} \ J(\theta) + \lambda \beta^T \beta \\ ② \ \underset{\theta}{\text{arg min}} \ J(\theta), \ \text{s.t.} \ \beta^T \beta \le S \end{cases}$$

Optimal solution β^*_{Rg} needs (necessary condition) $\lambda > 0$

$$\left[\lambda \left(\sum_j (\beta_{Rg})_j^2 - S \right) = 0 \right] \Rightarrow S = \sum_j (\beta_{Rg})_j^2$$

When $X^T X = I$,

$$S = \sum_j (\beta_{Rg})_j^2 = \frac{1}{(1+\lambda)^2} \sum_j (\beta_{OLS})_j^2$$

$$\Rightarrow S \propto \frac{1}{(1+\lambda)^2}$$

$$\lambda = \sqrt{\frac{\sum_j (\beta_{OLS})_j^2}{S}} - 1$$

http://stats.stackexchange.com/questions/190993/how-to-find-regression-coefficients-beta-in-ridge-regression

Extra: Intercept Term is usually not shrinked

- If the data is not centered, there exists bias term
  - http://stats.stackexchange.com/questions/86991/reason-for-not-shrinking-the-bias-intercept-term-in-regression

- We normally assume we centered x and y. If this is true, no need to have bias term, e.g. for lasso

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

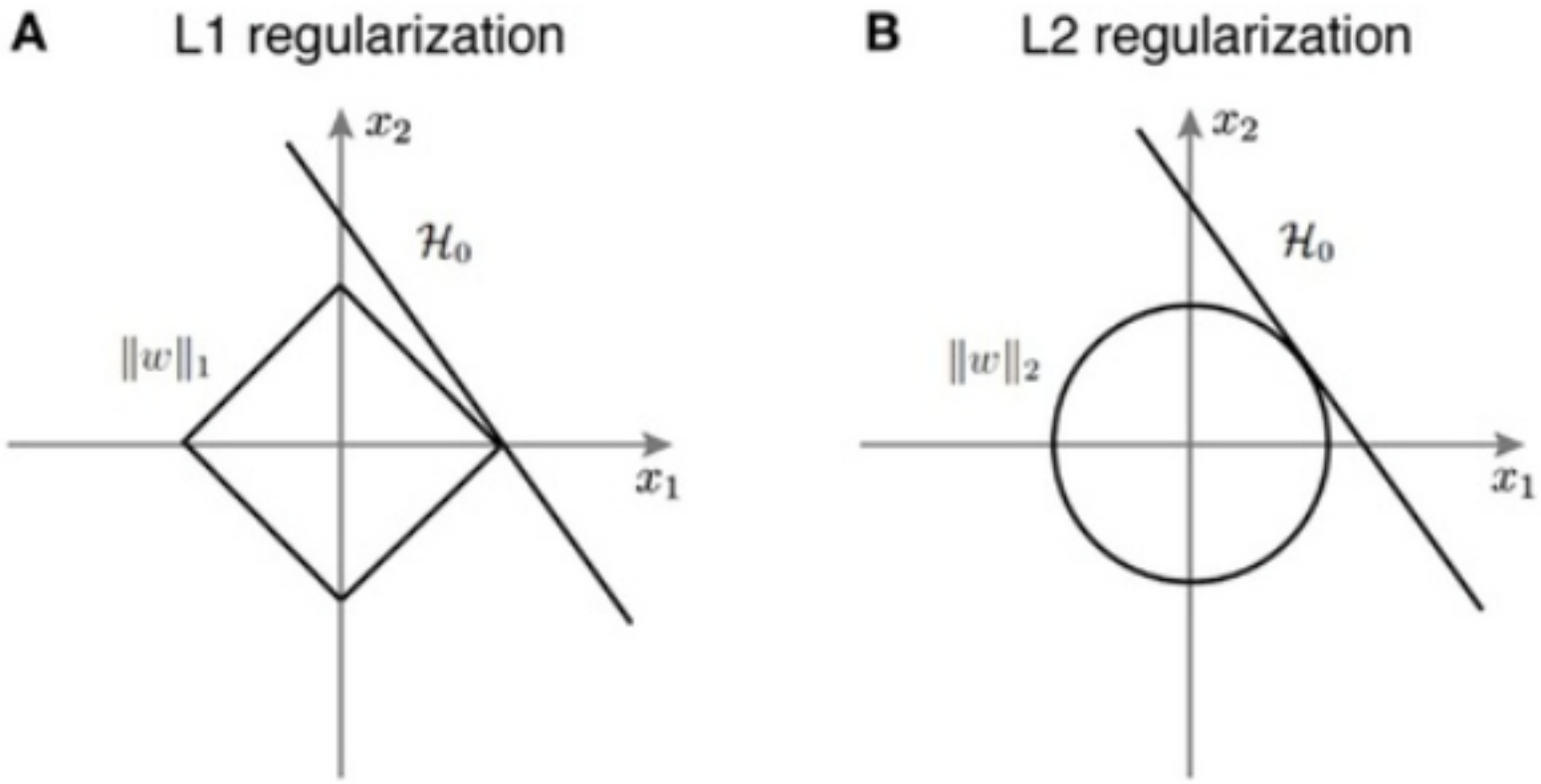
For ridge, in implementation just set the bias corresponding entry as 0 in the I-matrix

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1 \|\beta\|_1$$

for ridge $+\lambda \|\beta\|_2^2$

Extra Recap

- More about LR Model with Regularizations
  - Ridge Regression
  - Lasso Regression
    - Extra: how to perform training
  - Elastic net
    - Extra: how to perform training



due to the nature of L_1 norm, the viable solutions are limited to corners, which are on a few axis only - in the above case x1. Value of x2 = 0. This means that the solution has eliminated the role of x2, leading to sparsity

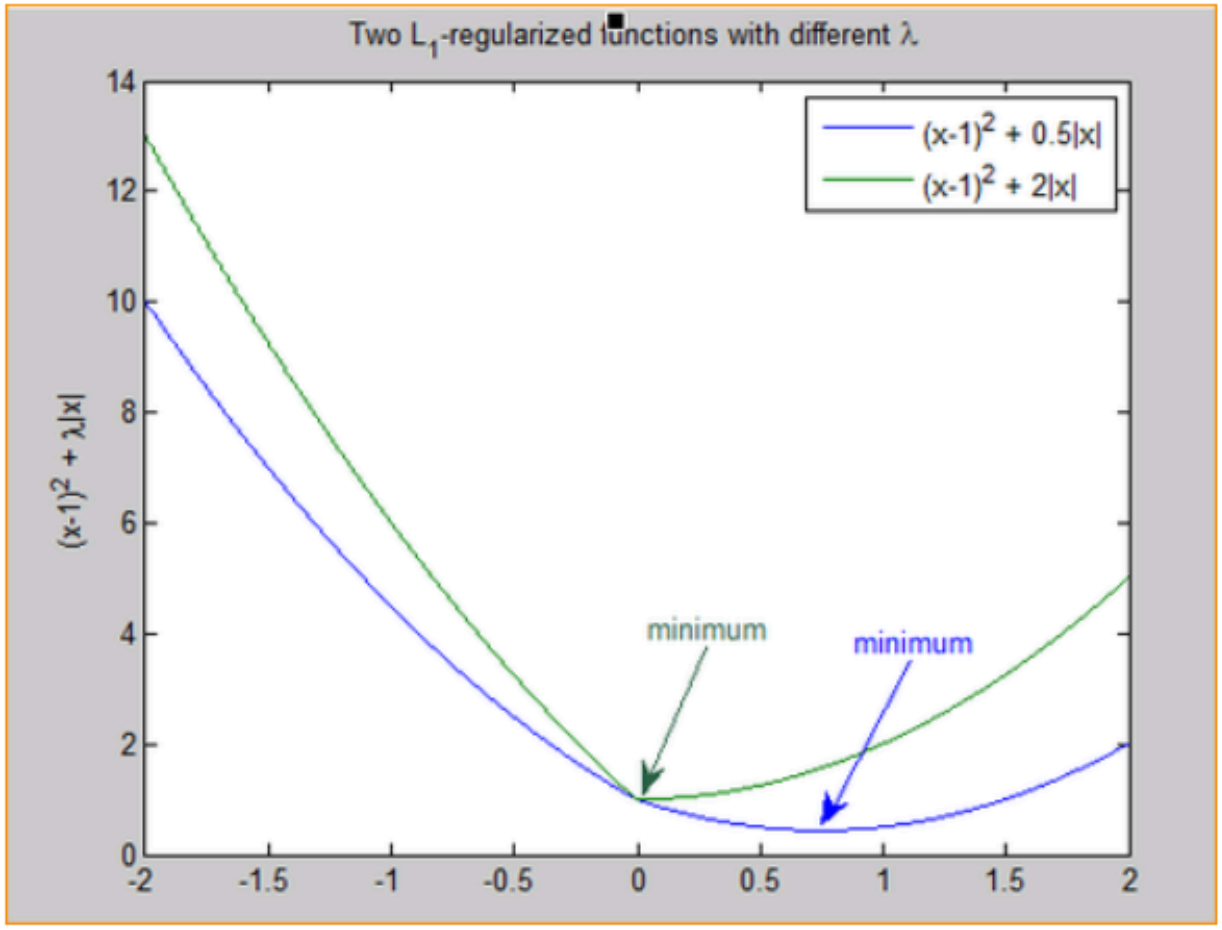
L_2-regularized loss function $F(x) = f(x) + \lambda \|x\|_2^2$ is smooth. This means that the optimum is the stationary point (0-derivative point). The stationary point of F can get very small when you increase λ, but still won't be 0 unless $f'(0) = 0$.

L_1-regularized loss function $F(x) = f(x) + \lambda \|x\|_1$ is non-smooth. It's not differentiable at 0. Optimization theory says that the optimum of a function is either the point with 0-derivative or one of the irregularities (corners, kinks, etc.). So, it's possible that the optimal point of F is 0 even if 0 isn't the stationary point of f. In fact, it would be 0 if λ is large enough (stronger regularization effect). Below is a graphical illustration.

> In multi-dimensional settings: if a feature is not important, the loss contributed by it is small and hence the (non-differentiable) regularization effect would turn it off.

L_1-regularized loss function $F(x) = f(x) + \lambda\|x\|_1$ is non-smooth. It's not differentiable at 0. Optimization theory says that the optimum of a function is either the point with 0-derivative or one of the irregularities (corners, kinks, etc.). So, it's possible that the optimal point of F is 0 even if 0 isn't the stationary point of f. In fact, it would be 0 if λ is large enough (stronger regularization effect). Below is a graphical illustration.

http://www.quora.com/What-is-the-difference-between-L1-and-L2-regularization



In mathematics, particularly in calculus, a stationary point or critical point of a differentiable function of one variable is a point of the domain of the function where the derivative is zero (equivalently, the slope of the graph at that point is zero).

How to train Parameter for Lasso

$$\hat{\beta}^{lasso} = \arg\min(y - X\beta)^T (y - X\beta)$$

$$\text{subject to } \sum \left|\beta_j\right| \leq s$$

- **ℓ_1-norm is non differentiable!**
  - cannot compute the gradient of the absolute value

$\Rightarrow$ **Directional derivatives** (or subgradient)

Here assume x and y have been centered (normally), therefore no bias term needed in above !

$$\text{RSS-Loss}(\lambda) = (y - X\beta)^T (y - X\beta) + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$= \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$= \left[\sum_{i=1}^{n} \left(y_i - x_{ij}\beta_j - x_{i\{-j\}}^T \beta_{-j} \right)^2 \right] + \lambda \sum_{j=1}^{p} |\beta_j|$$

if $\beta = (\beta_1, \beta_2, \beta_3)$

$\Rightarrow \beta_{-2} = (\beta_1, \beta_3)$

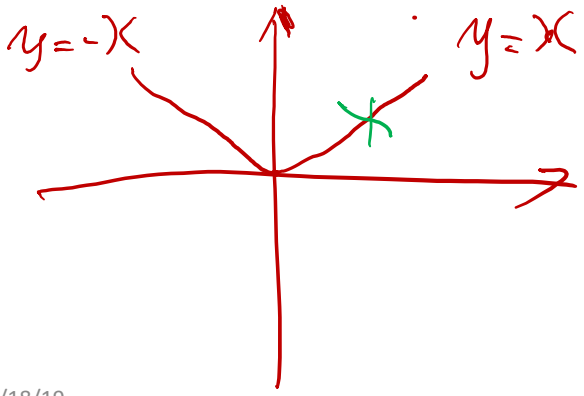
$$\Rightarrow \frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} 2\left(y_i - x_{ij}\beta_j - x_{i\{-j\}}^T \beta_{-j} \right)\left(-x_{ij} \right) + \lambda \frac{\partial}{\partial \beta_j} |\beta_j|$$

$$= \underbrace{2\sum_{i=1}^{n} x_{ij}^2 \beta_j}_{a_j} \underbrace{- 2\sum_{i=1}^{n} \left(y_i - x_{i\{-j\}}^T \beta_{\{-j\}}\right) x_{ij}}_{c_j} + \lambda \frac{\partial}{\partial \beta_j} |\beta_j|$$

$$= a_j \beta_j - c_j + \lambda \frac{\partial}{\partial \beta_j} |\beta_j| \overset{\text{set to}}{=\!=} 0$$

convext ⇒ unique

⊗ $y = f(x) = |x|$



$$\partial f(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \end{cases}$$

$$\frac{\partial \ell}{\partial \beta_j} = \begin{cases} a_j \beta_j - c_j - \lambda = 0, & \text{if } \beta_j < 0 \\ a_j \beta_j - c_j + \lambda, & \text{if } \beta_j > 0 \\ [a_j \beta_j - c_j - \lambda,\ a_j \beta_j - c_j + \lambda], & \text{if } \beta_j = 0 \end{cases}$$

Set to 0

$$\hat{\beta}_j = \begin{cases} \frac{c_j + \lambda}{a_j}, & \text{if } c_j + \lambda < 0 \Rightarrow c_j < -\lambda \\ \frac{c_j - \lambda}{a_j}, & \text{if } c_j > \lambda \\ 0, & \text{if } -\lambda \le c_j \le \lambda \end{cases}$$

Soft thresholding

We just need 0 in the region $[-c_j-\lambda, -c_j+\lambda]$ (subgradient calculus)

$$RSS(\lambda) = (y - x\beta)^T (y - x\beta) + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$= \sum_{i=1}^{n} (y_i - \underbrace{x_i^T \beta}_{\hat{y}}) + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$= \left[\sum_{i=1}^{n} (y_i - x_{ij}\beta_j - x_{i-j}^T \beta_{-j})^2 \right] + \left[\lambda \sum_{j=1}^{p} |\beta_j| \right]$$

Here, if $\beta = (\beta_1, \beta_2, \beta_3) \Rightarrow \beta_{-2} = (\beta_1, \beta_3)$

$$\frac{\partial}{\partial \beta_j} RSS(\lambda) = \sum_{i=1}^{n} 2(y_i - x_{ij}\beta_j - x_{i-j}^T \beta_j)(-x_{ij}) + \lambda \frac{\partial}{\partial \beta_j} (|\beta_1| + |\beta_2| + \cdots |\beta_p|)$$

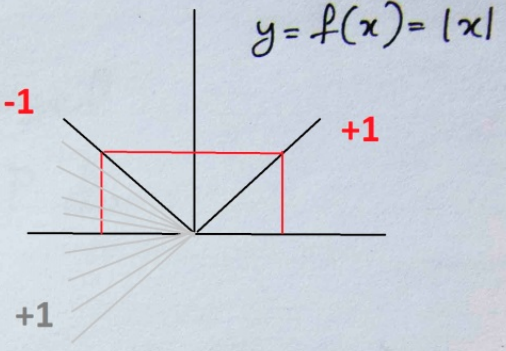
$$\doteq \underbrace{2 \sum_{i=1}^{n} x_{ij}^2}_{a_j} \beta_j - \underbrace{2 \sum_{i=1}^{n} (y_i - x_{i-j}^T \beta_{-j}) x_{ij}}_{c_j} + \lambda \frac{\partial}{\partial \beta_j} |\beta_j|$$

$$= a_j \beta_j - c_j + \lambda \frac{\partial}{\partial \beta_j} |\beta_j|$$

Sub differentials

$$f(x) = |x|$$

$$\partial f(x) = \begin{cases} \{-1\} & \text{if } x<0 \\ \{-1,1\} & \text{if } x=0 \\ \{+1\} & \text{if } x>0 \end{cases}$$



$$\frac{\partial}{\partial \beta_j} RSS(\lambda) = a_j \beta_j + c_j + \lambda \frac{\partial}{\partial \beta_j} |\beta_j|$$

$$= \begin{cases} \{a_j\beta_j - c_j - \lambda\} & \text{if } \beta_j<0 \\ \{-c_j-\lambda, -c_j+\lambda\} & \text{if } \beta_j=0 \\ \{a_j\beta_j - c_j + \lambda\} & \text{if } \beta_j>0 \end{cases}$$

$$\hat{\beta}_j = \begin{cases} (c_j+\lambda)/a_j & \text{if } c_j<-\lambda \\ 0 & \text{if } c_j \in [-\lambda,\lambda] \\ (c_j-\lambda)/a_j & \text{if } c_j>\lambda \end{cases}$$

When $\beta_j<0$: $a_j\beta_j - c_j - \lambda = 0$, $\beta_j = \frac{c_j+\lambda}{a_j}$

When $\beta_j>0$: $a_j\beta_j - c_j + \lambda = 0$, $\beta_j = \frac{c_j-\lambda}{a_j}$

Coordinate descent based Learning of Lasso

Coordinate descent (WIKI)➔ one does line search along one coordinate direction at the current point in each iteration.

One uses different coordinate directions cyclically throughout the procedure.

1. Initialize β
2. Repeat until converged
3. For $j = 1, 2, \ldots, p$ do

$$a_j = 2 \sum_{i=1}^{n} x_{ij}^2$$

$$c_j = 2 \sum_{i=1}^{n} x_{ij} \left(y_i - x_i^T \beta + x_{ij} \beta_j \right)$$

if $c_j < -\lambda$

$$\beta_j = (c_j + \lambda)/a_j$$

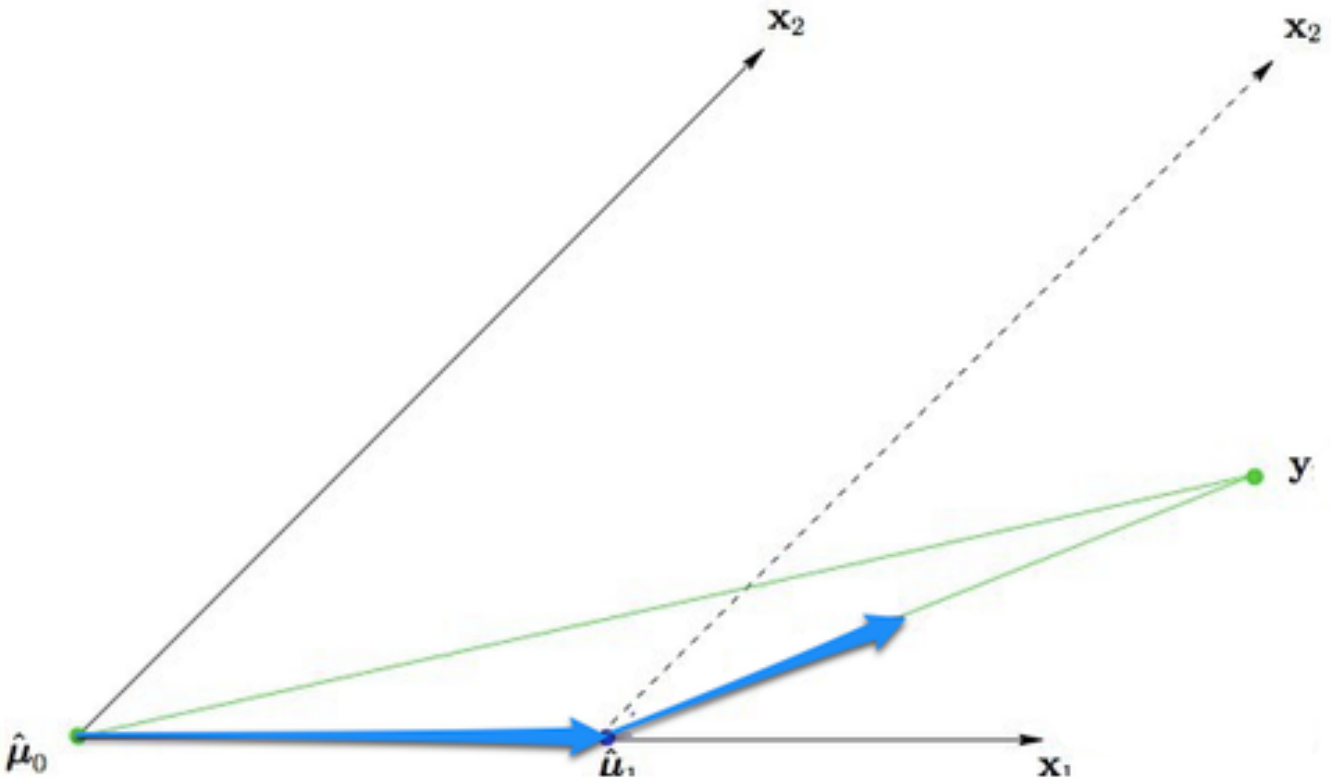
else if, $c_j > \lambda$

$$\beta_j = (c_j - \lambda)/a_j$$

else, soft-thresholding

$$\beta_j = 0$$

Least Angle Regression (LARS) (State-of-the-art LASSO solver)



http://statweb.stanford.edu/~tibs/ftp/lars.pdf

LARS: Least Angle Regression

- Starts like classic Forward Selection
  - Find predictor x_{j1} most correlated with the current residual
  - Make a step (epsilon) large enough until another predictor x_{j2} has as much correlation with the current residual
  - LARS – now step in the direction equiangular between two predictors until x_{j3} earns its way into the "correlated set"

Correlation: $$c(\mu) = X'(y - \mu)$$

Extra Recap

- More about LR Model with Regularizations
  - Ridge Regression
  - Lasso Regression
    - Extra: how to perform training
  - Elastic net
    - Extra: how to perform training

Naïve elastic net

- For any non negative fixed λ_1 and λ_2, naive elastic net criterion:

$$L(\lambda_1, \lambda_2, \beta) = |\mathbf{y} - \mathbf{X}\beta|^2 + \lambda_2|\beta|^2 + \lambda_1|\beta|_1,$$

$$|\beta|^2 = \sum_{j=1}^{p} \beta_j^2, \qquad |\beta|_1 = \sum_{j=1}^{p} |\beta_j|.$$

- The naive elastic net estimator $\hat{\beta}$ is the minimizer of equation

$$\hat{\beta} = \arg\min_{\beta} \{L(\lambda_1, \lambda_2, \beta)\}.$$

- Let

$$\alpha = \lambda_2/(\lambda_1 + \lambda_2)$$

$$\hat{\beta} = \arg\min_{\beta} |\mathbf{y} - \mathbf{X}\beta|^2, \qquad \text{subject to } (1-\alpha)\,|\beta|_1 + \alpha\,|\beta|^2 \leqslant t \text{ for some } t.$$

Naïve elastic net

- For any non negative fixed λ_1 and λ_2, naive elastic net criterion:

$$L(\lambda_1, \lambda_2, \beta) = |\mathbf{y} - \mathbf{X}\beta|^2 + \lambda_2|\beta|^2 + \lambda_1|\beta|_1,$$

$$|\beta|^2 = \sum_{j=1}^{p} \beta_j^2, \qquad |\beta|_1 = \sum_{j=1}^{p} |\beta_j|.$$

- The naive elastic net estimator is the minimizer of equation

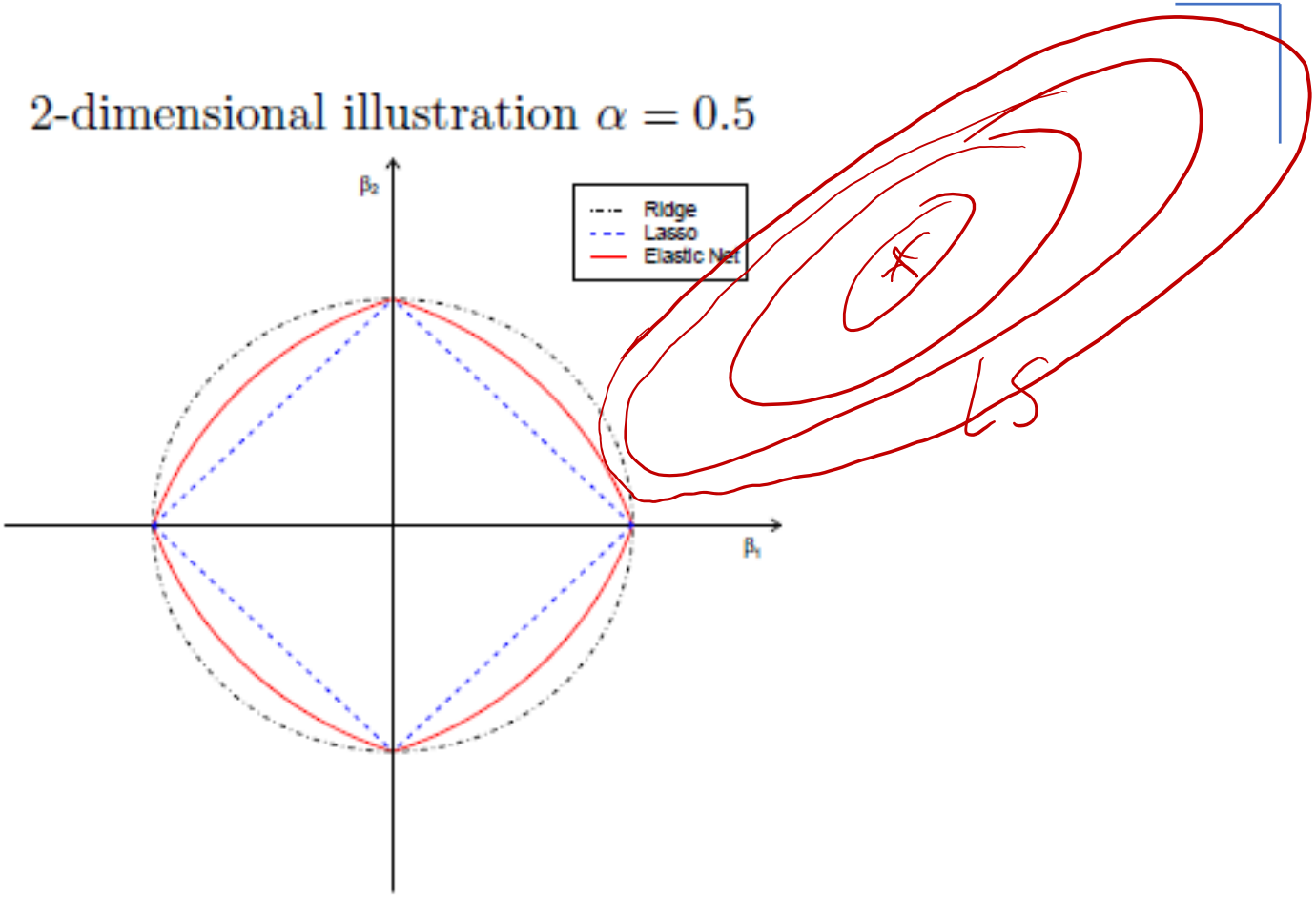
$$\hat{\beta} = \arg\min_{\beta} \{L(\lambda_1, \lambda_2, \beta)\}.$$

- Let

$$\alpha = \lambda_2/(\lambda_1 + \lambda_2) \qquad \frac{\lambda_2}{\lambda_1 + \lambda_2}$$

$$\hat{\beta} = \arg\min_{\beta} |\mathbf{y} - \mathbf{X}\beta|^2, \qquad \text{subject to } (1-\alpha)|\beta|_1 + \alpha|\beta|^2 \leqslant t \text{ for some } t.$$

Geometry of elastic net



Connecting LASSO and Naïve Elastic net

- Lemma: Given (λ_1, λ_2), define an artificial data set (y^*, X^*)

$$\mathbf{X}^*_{(n+p)\times p} = (1+\lambda_2)^{-1/2}\begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}\end{pmatrix}, \qquad \mathbf{y}^*_{(n+p)} = \begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix}.$$

Let $\gamma = \lambda_1/\sqrt{(1+\lambda_2)}$ and $\beta^* = \sqrt{(1+\lambda_2)}\beta$. Then the naïve elastic net criterion can be written as

$$L(\gamma, \beta) = L(\gamma, \beta^*) = \left|\mathbf{y}^* - \mathbf{X}^*\beta^*\right|^2 + \gamma\left|\beta^*\right|_1.$$

- Let, $\hat{\beta}^* = \arg\min_{\beta^*} L\{(\gamma, \beta^*)\};$

naive

- Then

$$\hat{\beta} = \frac{1}{\sqrt{(1+\lambda_2)}}\hat{\beta}^*.$$

Connecting LASSO and Naïve Elastic net

- Lemma: Given (λ_1, λ_2), define an artificial data set (y^*, X^*)

$$\mathbf{X}^*_{(n+p)\times p} = (1+\lambda_2)^{-1/2}\begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}\end{pmatrix}, \qquad \mathbf{y}^*_{(n+p)} = \begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix}.$$

Let $\gamma = \lambda_1/\sqrt{(1+\lambda_2)}$ and $\beta^* = \sqrt{(1+\lambda_2)}\beta$. Then the naïve elastic net criterion can be written as

$$L(\gamma, \beta) = L(\gamma, \beta^*) = \left|\mathbf{y}^* - \mathbf{X}^*\beta^*\right|^2 + \gamma\left|\beta^*\right|_1.$$

- Let, $\hat{\beta}^* = \arg\min_{\beta^*} L\{(\gamma, \beta^*)\};$
- Then

naive elastic

$$\hat{\beta} = \frac{1}{\sqrt{(1+\lambda_2)}}\hat{\beta}^*.$$

lasso augmented

$X\beta = Y$

$n \times p \quad p \times 1 \quad n \times 1$

$(n+p) \times p \quad p \times 1 \quad (n+p) \times 1$

$$Loss = \left| y^* - X^* \beta^* \right|^2 + r \left| \beta^* \right|_1$$

$$= \left| \begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{pmatrix} X \\ \sqrt{\lambda_2} I \end{pmatrix} \beta^* \right|^2 + r \left| \beta^* \right|_1$$

$(n+p) \times p$

$$= \left| \begin{pmatrix} y - X\beta^* \\ -\sqrt{\lambda_2}\beta^* \end{pmatrix} \right|^2 + r \left| \beta^* \right|_1$$

$$= (y - X\beta^*)^2 + \lambda_2 \beta^{*2} + r \left| \beta^* \right|_1$$

$$= (y - X\beta^*)^2 + \lambda_2 (1+\lambda_2)\beta^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}} \left| \sqrt{(1+\lambda_2)}\beta \right|_1$$

Advantage of Elastic net

- Native Elastic set can be converted to lasso with augmented data
- In the augmented formulation,
  - sample size n+p and X^* has rank p
  - ➔ can potentially select all the predictors
- Naïve elastic net can perform automatic variable selection like lasso

http://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Grouping Effect

Qualitatively speaking, a regression method exhibits the grouping effect if the regression coefficients of a group of highly correlated variables tend to be equal (up to a change of sign if negatively correlated). In particular, in the extreme situation where some variables are exactly identical, the regression method should assign identical coefficients to the identical variables.

If there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not care which one is selected.

uations. We illustrate our points by considering the gene selection problem in microarray data analysis. A typical microarray data set has many thousands of predictors (genes) and often fewer than 100 samples. For those genes sharing the same biological 'pathway', the correlations between them can be high (Segal and Conklin, 2003). We think of those genes as forming a group. The ideal gene selection method should be able to do two things: eliminate the trivial genes and automatically include whole groups into the model once one gene among them is selected ('grouped selection'). For this kind of $p \gg n$ and grouped variables situation, the lasso is not the ideal method, because it can only select at most n variables out of p candidates (Efron *et al.*, 2004), and it lacks the ability to reveal the grouping information. As for prediction per-

http://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Grouping Effect of Naïve Elastic net

$$\hat{\beta} = \arg\min_{\beta} |\mathbf{y} - \mathbf{X}\beta|^2 + \lambda J(\beta)$$

- Consider the following penalized regression model: Where J(.) positive for $\beta \neq 0$.

Lemma 2. Assume that $\mathbf{x}_i = \mathbf{x}_j$, $i, j \in \{1, \ldots, p\}$.

(a) If $J(\cdot)$ is strictly convex, then $\hat{\beta}_i = \hat{\beta}_j$, $\forall \lambda > 0$.
(b) If $J(\beta) = |\beta|_1$, then $\hat{\beta}_i \hat{\beta}_j \geqslant 0$ and $\hat{\beta}^*$ is another minimizer of equation (7), where

$$\hat{\beta}_k^* = \begin{cases} \hat{\beta}_k & \text{if } k \neq i \text{ and } k \neq j, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (s) & \text{if } k = i, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (1 - s) & \text{if } k = j, \end{cases}$$

for any $s \in [0, 1]$.

Lemma 2 shows a clear distinction between *strictly* convex penalty functions and the lasso penalty. Strict convexity guarantees the grouping effect in the extreme situation with identical predictors. In contrast the lasso does not even have a unique solution. The elastic net penalty with $\lambda_2 > 0$ is strictly convex, thus enjoying the property in assertion (1).

http://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Grouping Effect of Naïve Elastic net

$$\hat{\beta} = \arg\min_{\beta} |\mathbf{y} - \mathbf{X}\beta|^2 + \lambda J(\beta)$$

- Consider the following penalized regression model: Where J(.) positive for β ≠ 0.

Lemma 2. Assume that $\mathbf{x}_i = \mathbf{x}_j$, $i, j \in \{1, \ldots, p\}$.

(a) If $J(\cdot)$ is strictly convex, then $\hat{\beta}_i = \hat{\beta}_j$, $\forall \lambda > 0$.
(b) If $J(\beta) = |\beta|_1$, then $\hat{\beta}_i \hat{\beta}_j \geqslant 0$ and $\hat{\beta}^*$ is another minimizer of equation (7), where

$$\hat{\beta}_k^* = \begin{cases} \hat{\beta}_k & \text{if } k \neq i \text{ and } k \neq j, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (s) & \text{if } k = i, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (1 - s) & \text{if } k = j, \end{cases}$$

for any $s \in [0, 1]$.

Lasso does not provide a unique solution

Lemma 2 shows a clear distinction between *strictly* convex penalty functions and the lasso penalty. Strict convexity guarantees the grouping effect in the extreme situation with identical predictors. In contrast the lasso does not even have a unique solution. The elastic net penalty with $\lambda_2 > 0$ is strictly convex, thus enjoying the property in assertion (1).

http://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Grouping Effect of Naïve Elastic net

Theorem 1. Given data $(\mathbf{y}, \mathbf{X})$ and parameters (λ_1, λ_2), the response $\mathbf{y}$ is centred and the predictors $\mathbf{X}$ are standardized. Let $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ be the naïve elastic net estimate. Suppose that $\hat{\beta}_i(\lambda_1, \lambda_2)\,\hat{\beta}_j(\lambda_1, \lambda_2) > 0$. Define

$$D_{\lambda_1,\lambda_2}(i, j) = \frac{1}{|\mathbf{y}|_1}|\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)|;$$

then

$$D_{\lambda_1,\lambda_2}(i, j) \leqslant \frac{1}{\lambda_2}\sqrt{\{2(1-\rho)\}},$$

where $\rho = \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$, the sample correlation.

- D is the difference between the coefficient paths of predictors i and j.
- If x_i and x_j are high correlated ρ=1, this theorem provides a quantitative description for the grouping effect of Naive Elastic Net.

http://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Grouping Effect of Naïve Elastic net

Theorem 1. Given data $(\mathbf{y}, \mathbf{X})$ and parameters (λ_1, λ_2), the response $\mathbf{y}$ is centred and the predictors $\mathbf{X}$ are standardized. Let $\hat{\beta}(\lambda_1, \lambda_2)$ be the naïve elastic net estimate. Suppose that $\hat{\beta}_i(\lambda_1, \lambda_2)\,\hat{\beta}_j(\lambda_1, \lambda_2) > 0$. Define

$$D_{\lambda_1,\lambda_2}(i, j) = \frac{1}{|\mathbf{y}|_1}|\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)|;$$

then

$$D_{\lambda_1,\lambda_2}(i, j) \leqslant \frac{1}{\lambda_2}\sqrt{\{2(1-\rho)\}},$$

where $\rho = \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$, the sample correlation.

- D is the difference between the coefficient paths of predictors i and j.
- If x_i and x_j are high correlated ρ=1, this theorem provides a quantitative description for the grouping effect of Naive Elastic Net.

http://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Elastic Net

In the regression prediction setting, an accurate penalization method achieves good prediction performance through the bias–variance trade-off. The naïve elastic net estimator is a two-stage procedure: for each fixed λ_2 we first find the ridge regression coefficients, and then we do the lasso-type shrinkage along the lasso coefficient solution paths. It appears to incur a double amount of shrinkage. Double shrinkage does not help to reduce the variances much and introduces unnecessary extra bias, compared with pure lasso or ridge shrinkage. In the next section we improve the prediction performance of the naïve elastic net by correcting this double shrinkage.

- **Deficiency of the Naive Elastic Net:** Empirical evidence shows the Naive Elastic Net does not perform satisfactorily. The reason is that there are two shrinkage procedures (Ridge and LASSO) in it. Double shrinkage introduces unnecessary bias.
- Re-scaling of Naive Elastic Net gives better performance, yielding the Elastic Net solution:
- Reason: Undo shrinkage.

$$\hat{\beta}(\text{ENet}) = (1 + \lambda_2) \cdot \hat{\beta}(\text{Naive ENet})$$

http://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Elastic Net

3.2. *The elastic net estimate*

We follow the notation in Section 2.2. Given data $(\mathbf{y}, \mathbf{X})$, penalty parameter (λ_1, λ_2) and augmented data $(\mathbf{y}^*, \mathbf{X}^*)$, the naïve elastic net solves a lasso-type problem

$$\hat{\beta}^* = \arg\min_{\beta^*} |\mathbf{y}^* - \mathbf{X}^*\beta^*|^2 + \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}|\beta^*|_1. \quad (10)$$

The elastic net (corrected) estimates $\hat{\beta}$ are defined by

$$\hat{\beta}(\text{elastic net}) = \sqrt{(1+\lambda_2)}\hat{\beta}^*. \quad (11)$$

Recall that $\hat{\beta}(\text{naïve elastic net}) = \{1/\sqrt{(1+\lambda_2)}\}\hat{\beta}^*$; thus

$$\hat{\beta}(\text{elastic net}) = (1+\lambda_2)\hat{\beta}(\text{naïve elastic net}). \quad (12)$$

Hence the elastic net coefficient is a rescaled naïve elastic net coefficient.

Such a scaling transformation preserves the variable selection property of the naïve elastic net and is the simplest way to undo shrinkage. Hence all the good properties of the naïve elastic

Computation of elastic net

- First solve the Naive Elastic Net problem, then rescale it.
- For fixed λ_2, the Naive Elastic Net problem is equivalent to a LASSO problem, with a huge data matrix if p >> n
- LASSO already has an efficient solver called LARS (Least Angle Regression).
- ➔ LARS-EN algorithm.

http://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Elastic Net interpreted as a stabilized Lasso

Theorem 2. Given data $(\mathbf{y}, \mathbf{X})$ and (λ_1, λ_2), then the elastic net estimates $\hat{\boldsymbol{\beta}}$ are given by

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}} \left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda_2 \mathbf{I}}{1+\lambda_2} \right) \boldsymbol{\beta} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} + \lambda_1 |\boldsymbol{\beta}|_1. \tag{14}$$

It is easy to see that

$$\hat{\boldsymbol{\beta}}(\text{lasso}) = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}} (\mathbf{X}^{\mathrm{T}}\mathbf{X}) \boldsymbol{\beta} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} + \lambda_1 |\boldsymbol{\beta}|_1. \tag{15}$$

Hence theorem 2 interprets the elastic net as a stabilized version of the lasso. Note that $\hat{\Sigma} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$ is a sample version of the correlation matrix Σ and

$$\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda_2 \mathbf{I}}{1+\lambda_2} = (1-\gamma)\hat{\Sigma} + \gamma \mathbf{I}$$

with $\gamma = \lambda_2/(1+\lambda_2)$ shrinks $\hat{\Sigma}$ towards the identity matrix. Together equations (14) and (15) say that rescaling after the elastic net penalization is mathematically equivalent to replacing $\hat{\Sigma}$ with its shrunken version in the lasso. In linear discriminant analysis, the prediction accuracy can often be improved by replacing $\hat{\Sigma}$ by a shrunken estimate (Friedman, 1989; Hastie *et al.*, 2001). Likewise we improve the lasso by regularizing $\hat{\Sigma}$ in equation (15).

Extra Recap

- More about LR Model with Regularizations
  - Ridge Regression
  - Lasso Regression
    - Extra: how to perform training
  - Elastic net
    - Extra: how to perform training

References

- Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- Prof. Nando de Freitas's tutorial slide
- **Regularization and variable selection via the elastic net,** Hui Zou and Trevor Hastie*, Stanford University, USA*