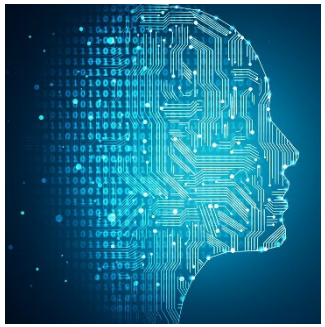
Machine Learning

Feature Selection and Regularization

Kevin Moon (kevin.moon@usu.edu)

STAT/CS 5810/6655



Feature Selection

- **Feature selection** is the problem of selecting a subset of features of a feature vector $\boldsymbol{x} = \left[x^{(1)}, \ldots, x^{(d)}\right]^T$ that are most relevant for a machine learning task (e.g. classification or regression)
- Motivations for feature selection:
  - Understanding/interpreting data
  - Improving computational efficiency
  - Improving performance (curse of dimensionality)
- Primary types of feature selection methods:
  - Filter methods
  - Wrapper methods
  - Embedded methods

Curse of dimensionality

- Data analysis becomes more difficult (statistically and computationally) as the dimension increases
- **Example**: a classification problem where

$$\boldsymbol{X}|Y = 1 \sim \mathcal{N}(\mu_1, I), \qquad \mu_1 = [1,0,\dots,0]^T$$
$$\boldsymbol{X}|Y = -1 \sim \mathcal{N}(\mu_{-1}, I), \qquad \mu_{-1} = [-1,0,\dots,0]^T$$

- Only the first feature is relevant for classification
- As $d \to \infty$, the distance between two random points in the same class has the *same distribution* as the distance between two random points in opposite classes
- Feature selection can significantly improve performance when only a few features are relevant

Curse of dimensionality

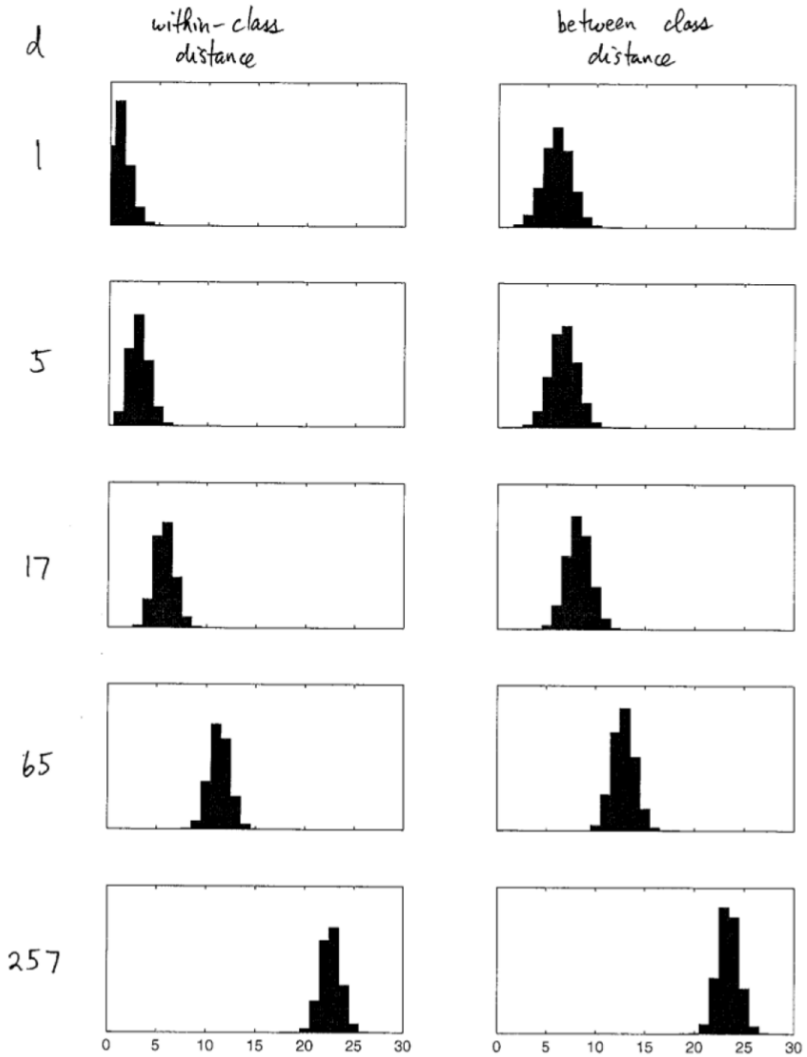


Figure 1: "Within Class" and "Between Class" distances as the dimension increases .

Filter Methods

- **Basic idea**: sort features by estimated relevance, take the top k features (k is the desired number)
- Consider a supervised learning problem with data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$
  - Common relevance measure for <u>classification</u>:

$$\left|t^{(j)}\right| = \frac{\left|\overline{x_1}^{(j)} - \overline{x_{-1}}^{(j)}\right|}{s_j/\sqrt{n}}$$

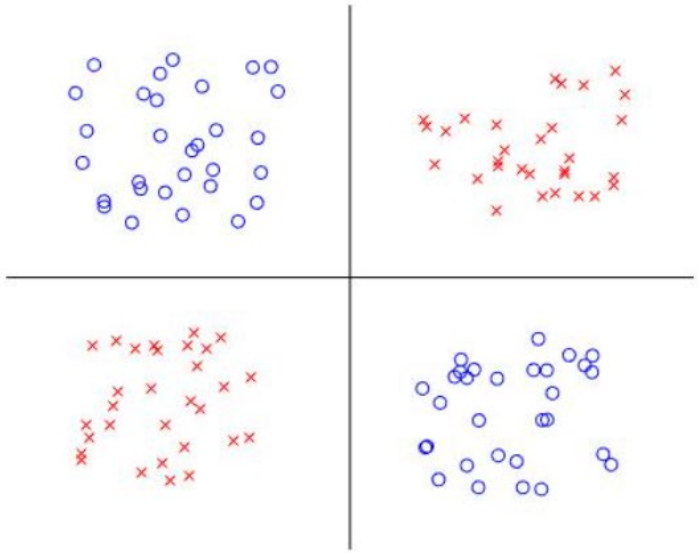
Where $\overline{x_m}^{(j)}$ = sample mean of $\left\{x_i^{(j)} \middle| y_i = m\right\}$

and s_j = pooled sample standard deviation of $\left\{x_i^{(j)}\right\}$.

  - Common relevance measure for <u>regression</u> is the correlation coefficient between Y and $X^{(j)}$

Filter Methods

- **Advantage**: Fast
- **Disadvantage**: the individually top k features are generally not the best collective k features



  - Each feature is useless individually, but collectively they can perfectly classify the data

Wrapper Methods

Three basic ingredients:

1. A machine learning algorithm
   - **Examples**: LDA, logistic regression, SVM, kernel ridge regression
2. A method for evaluating the performance of the algorithm when trained on a subset of features
   - **Examples**: Holdout, cross-validation
3. A strategy for searching through subsets of features
   - **Examples**: forward selection, backward elimination

- Wrapper methods derive their name from the fact that they wrap around the basic ML algorithm running it many times on different subsets of features

Forward selection and backward elimination

Both methods are greedy

- Forward selection
  - Start with $S = \{\}$
  - Given a subset S, increase the subset to $S \cup \{j\}$ where j gives the biggest increase in performance
- Backward elimination
  - Start with $S = \{1,2,\ldots,d\}$
  - Given a subset S, decrease the subset to $S \setminus \{j\}$, where $j \in S$ gives the smallest decrease in performance

Wrapper methods

- **Advantage**: Captures feature interactions
- **Disadvantages**: slow, not necessarily optimal (generally ok for just prediction, but less ok for interpretation)

- Can also modify the wrapper approach to use mutual information as the measure of performance (no training required)

Embedded Methods

- Embedded methods perform feature selection and function estimation (e.g. classification or regression) simultaneously
- We'll focus on the Lasso (least absolute shrinkage and selection operator)

Lasso Regression

- A regression method that solves:

$$\min_{\boldsymbol{w},b} \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i - b)^2 + \lambda\|\boldsymbol{w}\|_1$$

  - $\|\boldsymbol{w}\|_1 = \sum_{j=1}^{d}\left|w^{(j)}\right|$ is called the ℓ_1 norm
  - This is least squares linear regression with ℓ_1 regularization
  - Can generalize to $0 < p < \infty$:

$$\|\boldsymbol{w}\|_p = \left(\sum_{j=1}^{d}\left|w^{(j)}\right|^p\right)^{\frac{1}{p}}$$

  - This is a true norm for $p \geq 1$ (triangle inequality fails for $p < 1$)

Lasso Regression

- From previous lectures, the optimal b is

$$\hat{b} = \bar{y} - \hat{\boldsymbol{w}}^T \bar{\boldsymbol{x}}$$

$$\hat{w} = \arg\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} (\tilde{y}_i - \boldsymbol{w}^T \tilde{\boldsymbol{x}}_i)^2 + \lambda \|\boldsymbol{w}\|_1$$

$$\tilde{y}_i = y_i - \bar{y}, \qquad \tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}$$

- Matrix-vector form:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \frac{1}{n} \left\| \tilde{y} - \tilde{X}\boldsymbol{w} \right\|_2^2 + \lambda \|\boldsymbol{w}\|_1$$

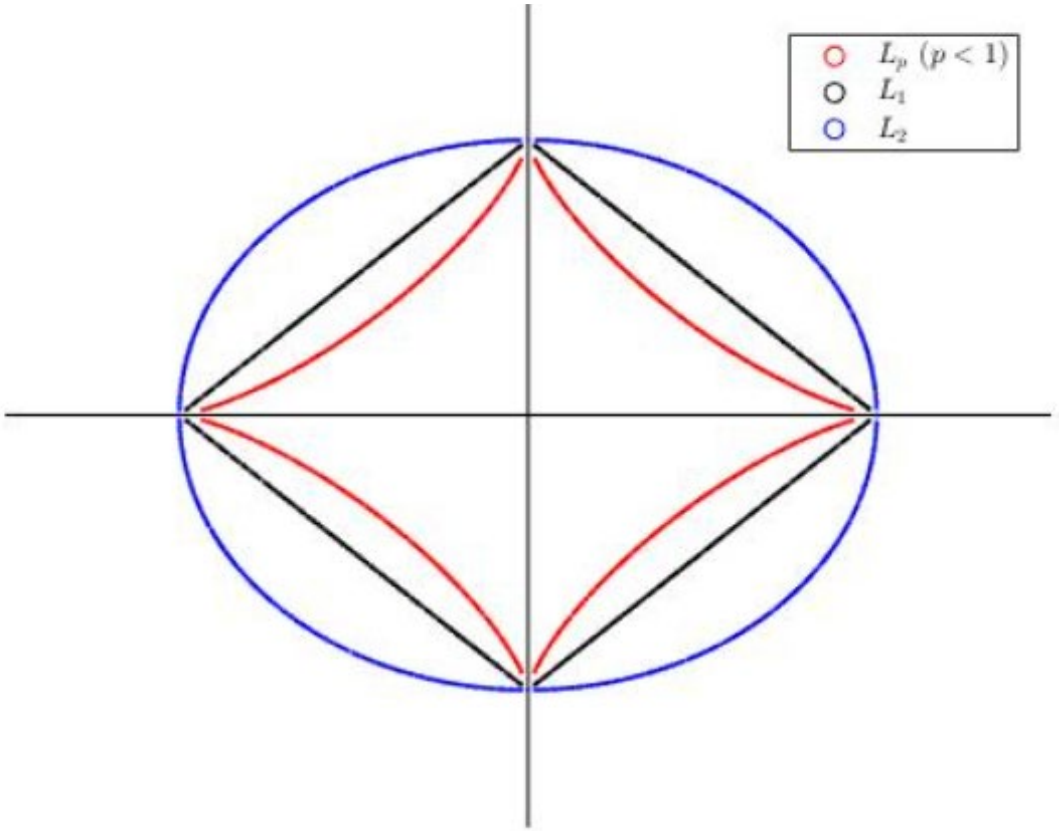
or (from Lagrange multiplier theory):

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \frac{1}{n} \left\| \tilde{y} - \tilde{X}\boldsymbol{w} \right\|_2^2$$

$$s.t. \ \|\boldsymbol{w}\|_1 \leq s$$

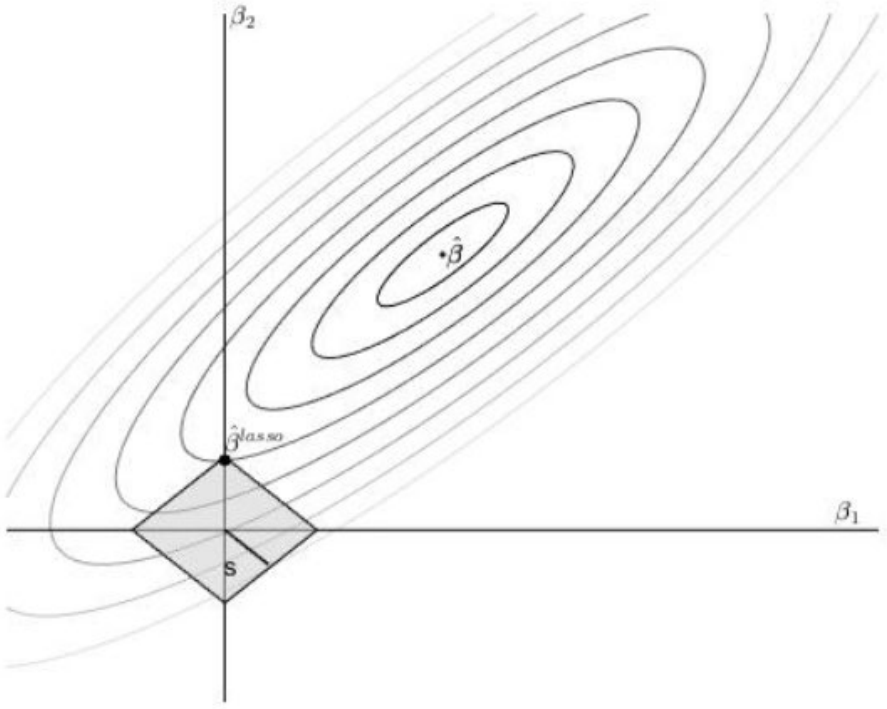
Lasso Regression

- Key observation: $\widehat{\boldsymbol{w}}$ is <u>sparse</u>
  - $\Rightarrow$ Lasso automatically selects the relevant features
- One explanation: shape of the ℓ_1 ball $\{\boldsymbol{w} | \|\boldsymbol{w}\|_1 \leq s\}$



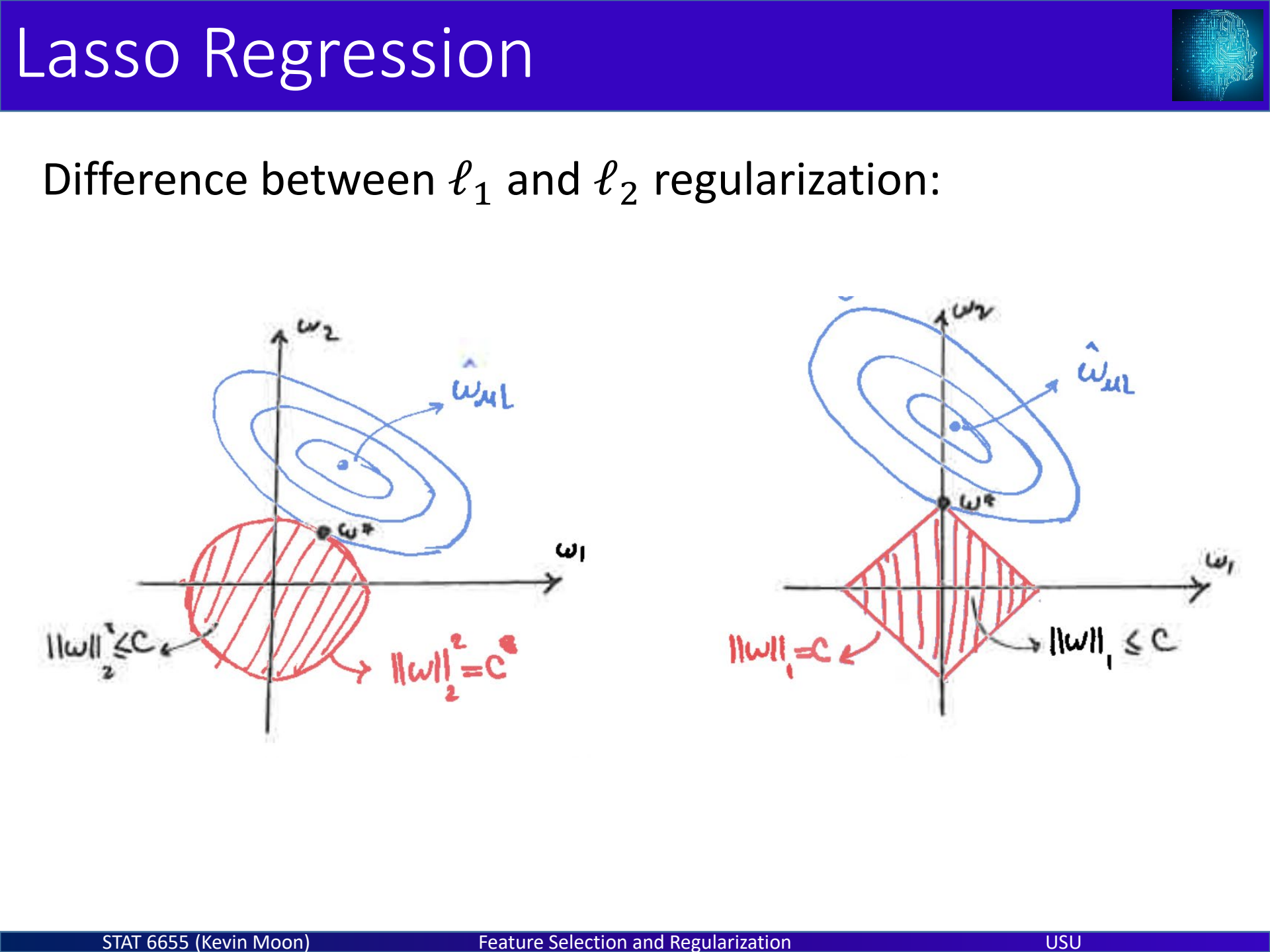
Lasso Regression

- Meanwhile, the set $\left\{\boldsymbol{w}\middle|\left\|\tilde{y}-\tilde{X}\boldsymbol{w}\right\|_2^2=c\right\}$ is an ellipse
- The minimum c value, subject to the constraint $\|\boldsymbol{w}\|_1 \leq s$, typically occurs along the $\boldsymbol{w}$ axes (thus some/many of the components will be zero)
- Smaller s is equivalent to larger λ and results in sparser $\boldsymbol{w}$



ℓ_1 norm and elliptical contours

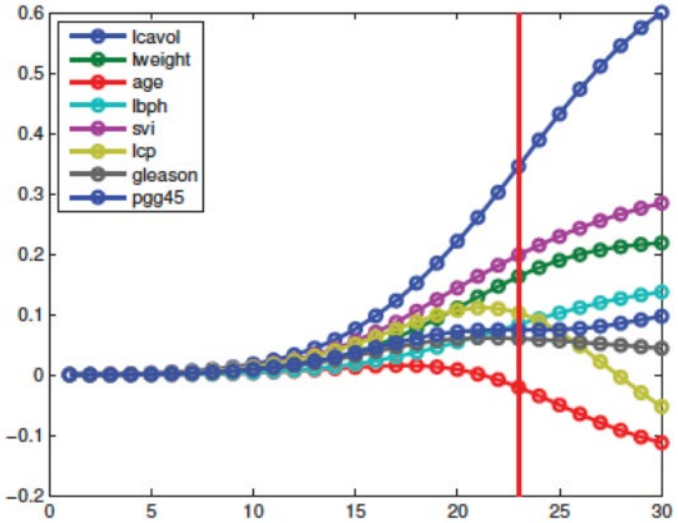
Lasso Regression



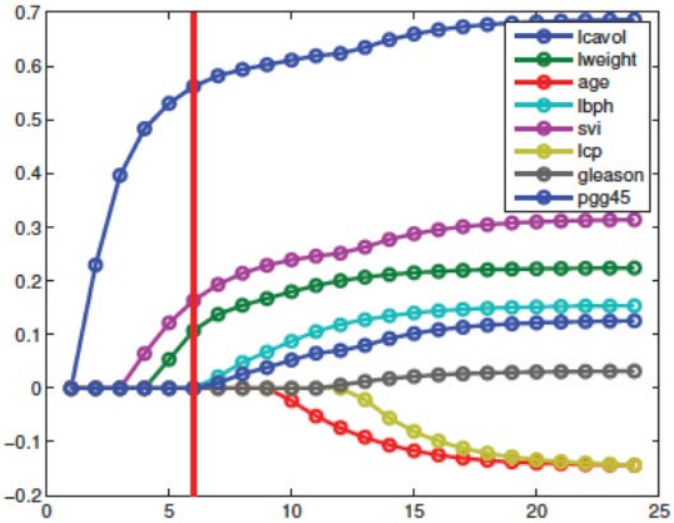
Difference between ℓ_1 and ℓ_2 regularization:

More on Regularization

- A significant disadvantage of the ℓ_1 penalty relative to ℓ_2 is that the problem cannot be kernelized
- In ℓ_2 regularization, all the coefficients are zero when $\lambda = \infty$ ($s = 0$). But for any finite values of λ, all coefficients are nonzero and increase in magnitude as λ decreases (s increases; (a))
- In ℓ_1 regularization, all the coefficients are zero when $\lambda = \infty$ ($s = 0$). As we decrease λ (increase s), the w's gradually turn on. If s is sufficiently small, the solution is sparse (b).



(a)



(b)

Further Reading

- ISL Chapter 6
- ESL Sections 3.4