UVA CS 4501 - 001 / 6501 – 007 Introduction to Machine Learning and Data Mining

Lecture 6: Regression Models with Regularization

Yanjun Qi / Jane

University of Virginia
Department of
Computer Science

Last Lecture Recap

- Linear model is an approximation
- Three ways to moving beyond linearity
  - LR with non-linear basis functions
  - Locally weighted linear regression
  - Regression trees and Multilinear Interpolation (later)

(1) LR with non-linear basis functions

- LR does not mean we can only deal with linear relationships

$$y = \theta_0 + \sum_{j=1}^{m} \theta_j \phi(x) = \theta^T \phi(x)$$

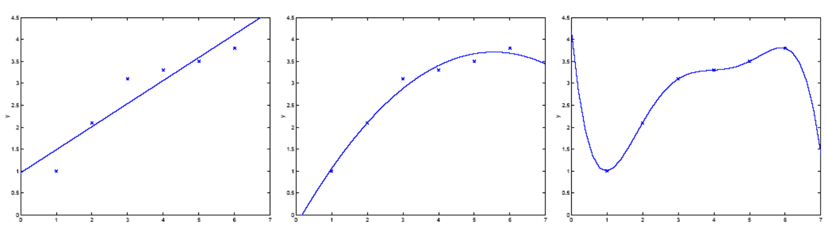
- We are free to design (non-linear) features (e.g., basis function derived) under LR

where the $\phi_j(x)$ are fixed basis functions (also define $\phi_0(x) = 1$).

- E.g.: polynomial regression:

$$\phi(x) := \left[1, x, x^2, x^3\right]$$

(1) LR With basis functions e.g. **polynomial regression** Issue: Overfitting OR underfitting



$y = \theta_0 + \theta_1 x$ $\quad$ $y = \theta_0 + \theta_1 x + \theta_2 x^2$ $\quad$ $y = \sum_{j=0}^{5} \theta_j x^j$

Generalisation: learn function / hypothesis from past data in order to "explain", "predict", "model" or "control" new data examples

K-fold Cross Validation !!!!

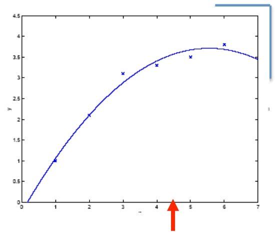
(2) Locally weighted linear regression

- The algorithm:

Instead of minimizing $J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta - y_i)^2$

now we fit ϑ to minimize $J(\theta) = \frac{1}{2}\sum_{i=1}^{n}w_i(\mathbf{x}_i^T\theta - y_i)^2$

Where do w_i's come from? $w_i = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x})^2}{2\tau^2}\right)$



  - where $\mathbf{x}$ is the query point for which we'd like to know its corresponding $\mathbf{y}$

→ Essentially we put higher weights on (errors on) training examples that are close to the query point (than those that are further away from the query)

Yanjun Qi / UVA CS 4501-01-6501-07

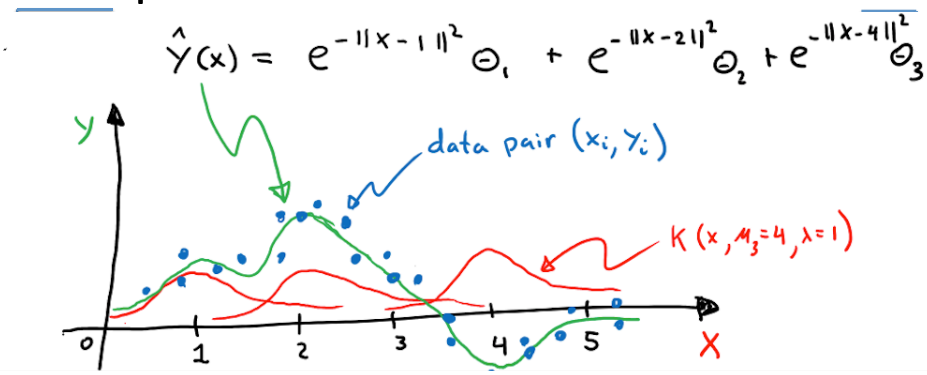
Parametric vs. non-parametric

- Locally weighted linear regression is a **non-parametric** algorithm.
- The (unweighted) linear regression algorithm that we saw earlier is known as a **parametric** learning algorithm
  - because it has a fixed, finite number of parameters (the *theta*), which are fit to the data;
  - Once we've fit the *theta* and stored them away, we no longer need to keep the training data around to make future predictions.
  - In contrast, to make predictions using locally weighted linear regression, we need to keep the entire training set around.
- The term "non-parametric" (roughly) refers to the fact that the amount of stuff we need to keep in order to represent the hypothesis grows with linearly the size of the training set.

Today

- ❑ A bit more about Linear Regression Extension
  - ❑ Linear regression with predefined RBF basis
  - ❑ Locally weighted regression
- ❑ An Exemplar Application of Regression
- ❑ Linear Regression Models with Regularization

(1) Linear regression with predefined RBF basis functions

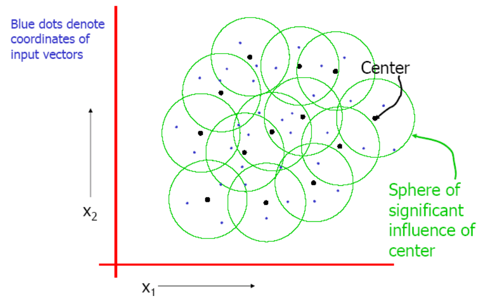


$$\varphi(x) := [1, k(x,1,1), k(x,2,1), k(x,4,1)]$$

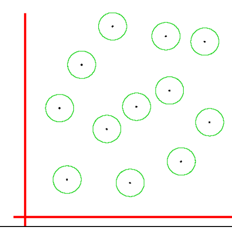
Dr. Nando de Freitas's tutorial slide

Issue: Choices of Basis Functions: ➔ Good and Bad RBFs

- A good 2D RBF



- A bad 2D RBFs



(2) Locally weighted regression

- *aka* locally weighted regression, locally linear regression, LOESS, …

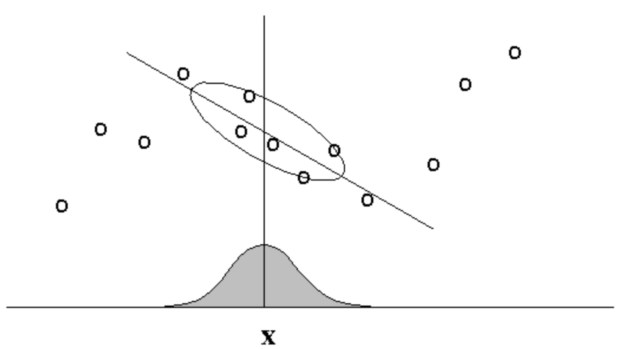


Figure 2: In locally weighted regression, points are weighted by proximity to the current x in question using a kernel. A regression is then computed using the weighted points.

(2) Locally weighted linear regression

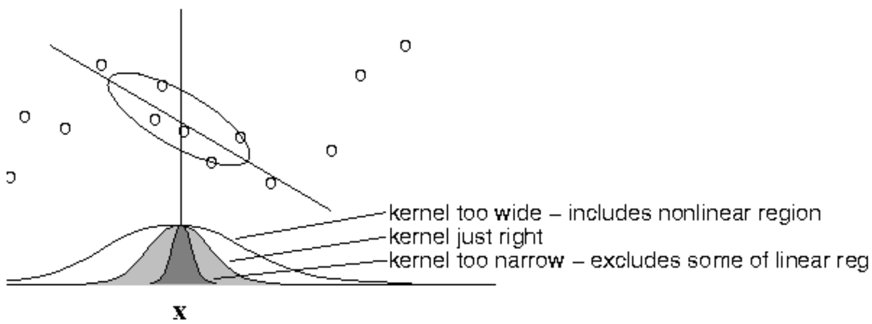


Figure 3: The estimator variance is minimized when the kernel includes as many training points as can be accommodated by the model. Here the linear LOESS model is shown. Too large a kernel includes points that degrade the fit; too small a kernel neglects points that increase confidence in the fit.

(2) Locally weighted linear regression

e.g. when for only one feature variable

- Separate weighted least squares **at each target point x_0:**

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0)x_i]^2$$

$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$$

- $b(x)^T=(1,x)$; B: Nx2 regression matrix with i-th row $b(x)^T$; $W_{N\times N}(x_0) = diag\left(K_\lambda(x_0, x_i)\right), i = 1,\ldots,N$

$$\hat{f}(x_0) = b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0) y$$

versus

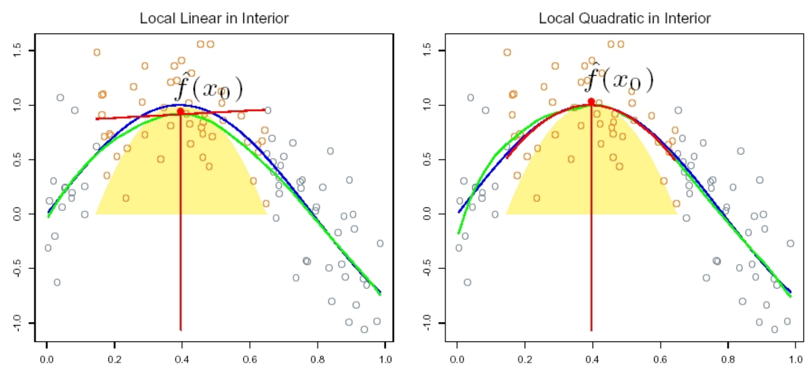
LR $\hat{f}(x_q) = (x_q)^T \theta^* = (x_q)^T \left(X^T X\right)^{-1} X^T \vec{y}$

(2). One More ➔ Local Weighted Polynomial Regression

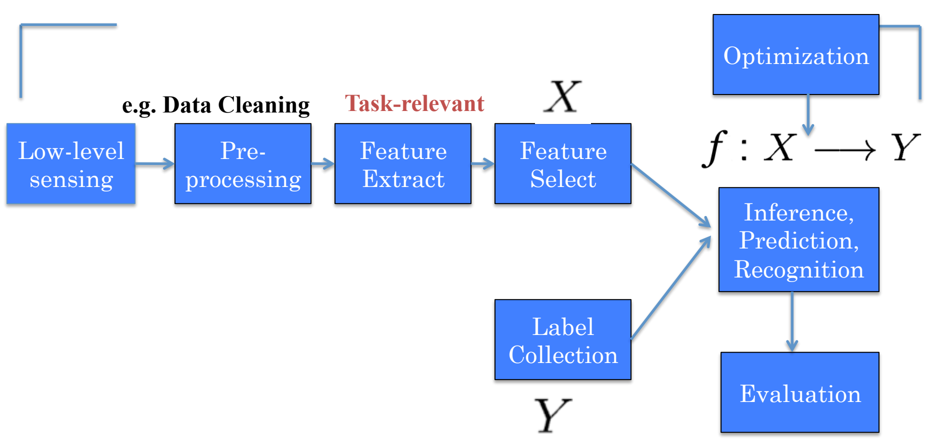
- Local polynomial fits of any degree d

$$\min_{\alpha(x_0),\beta_j(x_0),j=1,\ldots,d} \sum_{i=1}^{N} K_\lambda(x_0,x_i)\left[y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0)x_i^j\right]^2$$

$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \sum_{j=1}^{d} \hat{\beta}_j(x_0)x_0^j$$



TYPICAL MACHINE LEARNING SYSTEM



Today

- ❑ A bit more about Linear Regression Extension
  - ❑ Linear regression with predefined RBF basis
  - ❑ Locally weighted regression
- ❑ An Exemplar Application of Regression
- ❑ Linear Regression Models with Regularization

e.g. A Practical Application of Regression Model

Movie Reviews and Revenues: An Experiment in Text Regression*

Mahesh Joshi Dipanjan Das Kevin Gimpel Noah A. Smith
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{maheshj,dipanjan,kgimpel,nasmith}@cs.cmu.edu

Abstract

We consider the problem of predicting a movie's opening weekend revenue. Previous work on this problem has used metadata about a movie—e.g., its genre, MPAA rating, and cast—with very limited work making use of text *about* the movie. In this paper, we use the text of film critics' reviews from several sources to predict opening weekend revenue. We describe a new dataset pairing movie reviews with metadata and revenue data, and show that review text can substitute for metadata, and even improve over it, for prediction.

I. The Story in Short

- Use metadata and critics' reviews to predict opening weekend revenues of movies
- Feature analysis shows what aspects of reviews predict box office success

II. Data

- 1718 Movies, released 2005-2009
- Metadata (genre, rating, running time, actors, director, etc.): *www.metacritic.com*
- Critics' reviews (~7K): Austin Chronicle, Boston Globe, Entertainment Weekly, LA Times, NY Times, Variety, Village Voice
- Opening weekend revenues and number of opening screens: *www.the-numbers.com*

Movie Reviews and Revenues: An Experiment in Text Regression, Proceedings of HLT '10 Human Language Technologies:

III. Model

- Linear regression with the elastic net (Zou and Hastie, 2005)

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}=(\beta_0,\boldsymbol{\beta})}{\operatorname{argmin}} \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - (\beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta}) \right)^2 + \lambda P(\boldsymbol{\beta})$$

$$P(\boldsymbol{\beta}) = \sum_{j=1}^{p} \left(\tfrac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right)$$

Use linear regression to directly predict the opening weekend gross earnings, denoted y, based on features x extracted from the movie metadata and/or the text of the reviews.

Movie Reviews and Revenues: An Experiment in Text Regression, Proceedings of HLT '10 Human Language Technologies:

e.g. counts of a ngram in the text

IV. Features

I	Lexical n-grams (1,2,3)
II	Part-of-speech n-grams (1,2,3)
III	Dependency relations (nsubj,advmod,…)
Meta	U.S. origin, running time, budget (log), # of opening screens, genre, MPAA rating, holiday release (summer, Christmas, Memorial day,…), star power (Oscar winners, high-grossing actors)

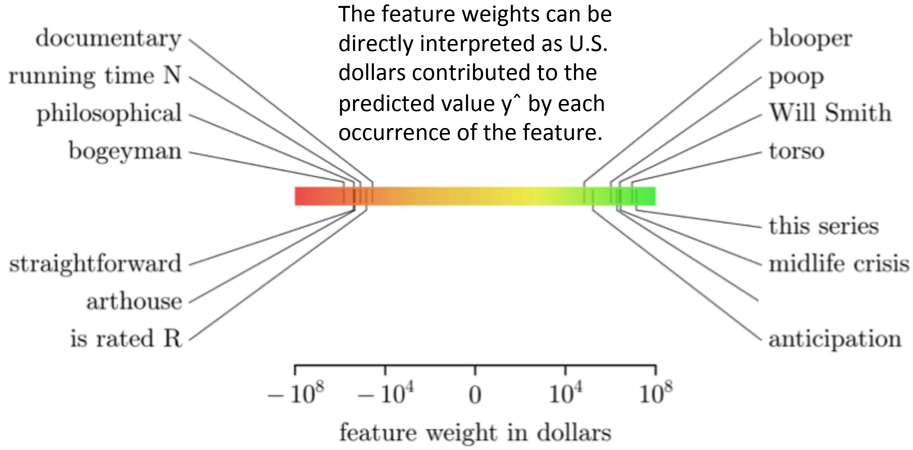
HW2

VIII. Get the Data!

www.ark.cs.cmu.edu/movie$-data

V. What May Have Brought You to movies

The feature weights can be directly interpreted as U.S. dollars contributed to the predicted value y^ by each occurrence of the feature.



Today

- ❑ A bit more about Linear Regression Extension
  - ❑ Linear regression with predefined RBF basis
  - ❑ Locally weighted regression
- ❑ An Exemplar Application of Regression
- ❑ Linear Regression Model with Regularizations
  - ❑ Ridge Regression
  - ❑ Lasso Regression

Yanjun Qi / UVA CS 4501-01-6501-07

Review: Vector norms

A norm of a vector ||x|| is informally a measure of the "length" of the vector.

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

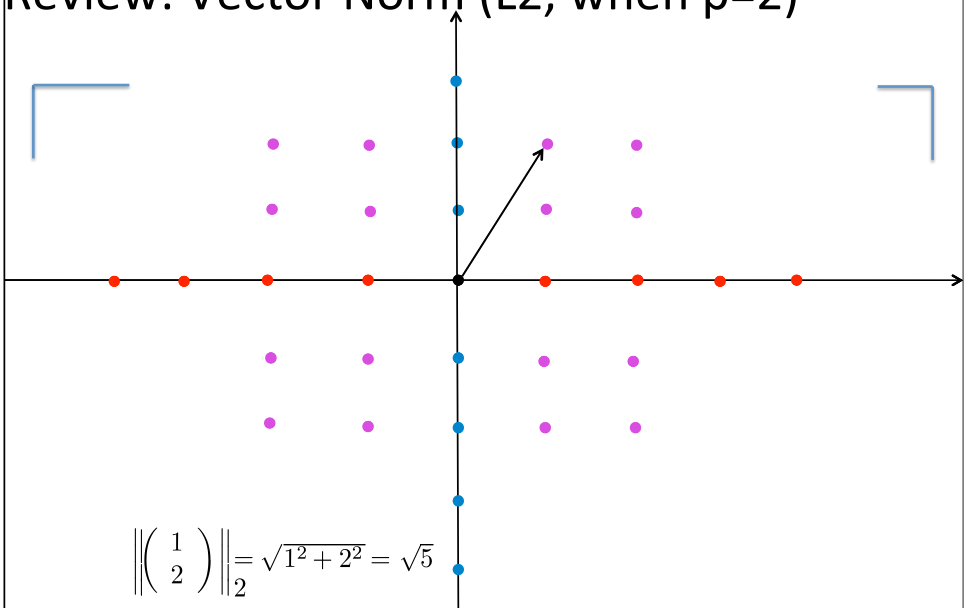
– Common norms: L_1, L_2 (Euclidean)

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \qquad \|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

– $L_{infinity}$

$$\|x\|_\infty = \max_i |x_i|$$

Review: Vector Norm (L2, when p=2)



$$\left\| \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right\|_2 = \sqrt{1^2 + 2^2} = \sqrt{5}$$

Review: Normal equation for LR

- Write the cost function in matrix form:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta - y_i)^2$$
$$= \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y})$$
$$= \frac{1}{2}\left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T\vec{y}\right)$$

$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1^T & -- \\ -- & \mathbf{x}_2^T & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n^T & -- \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

To minimize *J(θ)*, take derivative and set to zero:

$$\Rightarrow X^T X\theta = X^T\vec{y}$$

The normal equations

$$\Downarrow$$

$$\theta^* = \left(X^T X\right)^{-1} X^T \vec{y}$$

Assume that X^TX is invertible

(1) Ridge Regression / L2

- If not invertible, a solution is to add a small element to diagonal

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$$ Basic Model,

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

- The ridge estimator is solution from

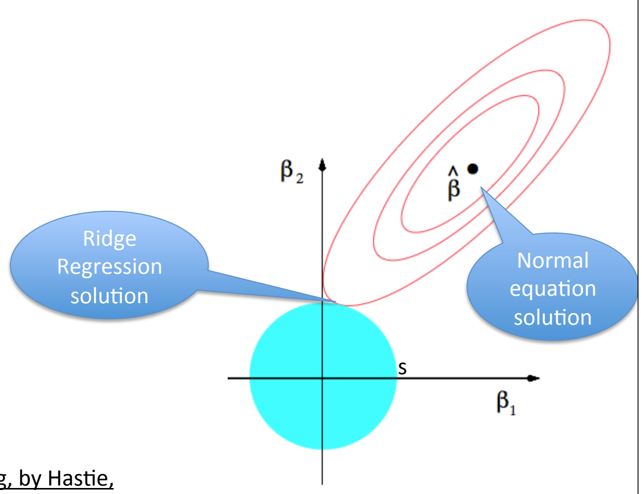
HW2

$$\hat{\beta}^{ridge} = \text{argmin}(y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta$$

- Equivalently

$$\hat{\beta}^{ridge} = \arg\min (y - X\beta)^T (y - X\beta)$$
$$\text{subject to } \sum \beta_j^2 \le s$$

Objective Function's Contour lines from Ridge Regression



Elements of Statistical Learning, by Hastie, Tibshirani and Friedman

(1) Ridge Regression / L2

- The parameter $\lambda > 0$ penalizes β_j proportional to its size β_j^2
- Solution is $\hat{\beta}_\lambda = (X^T X + \lambda I)^{-1} X^T y$
- where I is the identity matrix.
- Note $\lambda = 0$ gives the least squares estimator;
- if $\lambda \to \infty$, then $\hat{\beta} \to 0$

(2) Lasso (least absolute shrinkage and selection operator) / L1

- The lasso is a shrinkage method like ridge, but acts in a nonlinear manner on the outcome y.
- The lasso is defined by

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

$$\text{subject to } \sum |\beta_j| \le s$$

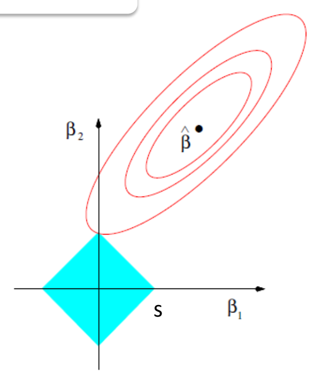
Lasso (least absolute shrinkage and selection operator)

- Notice that ridge penalty $\sum \beta_j^2$ is replaced by $\sum |\beta_j|$
- Due to the nature of the constraint, if tuning parameter is chosen small enough, then the lasso will set some coefficients exactly to zero.

Lasso (least absolute shrinkage and selection

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

- Suppose in 2 dimension
- $\beta = (\beta_1, \beta_2)$
- $|\beta_1| + |\beta_2| = \text{const}$
- $|\beta_1| + |-\beta_2| = \text{const}$
- $|-\beta_1| + |\beta_2| = \text{const}$
- $|-\beta_1| + |-\beta_2| = \text{const}$



Elements of Statistical Learning, by Hastie, Tibshirani and Friedman

Elements of Statistical Learning, by Hastie, Tibshirani and Friedman

Linear Methods for Regression

(3) A family of shrinkage estimators

$$\beta = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$$
$$\text{subject to} \quad \sum \left|\beta_j\right|^q \leq s$$

- for q >=0, contours of constant value of $\sum_j |\beta_j|^q$ are shown for the case of two inputs.

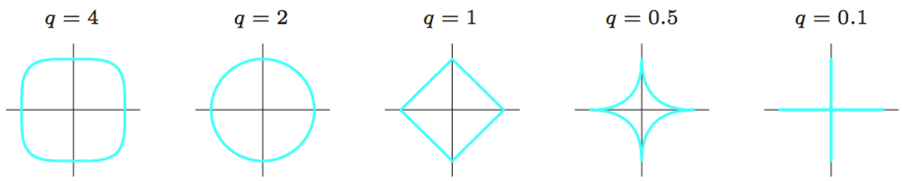


FIGURE 3.12. *Contours of constant value of* $\sum_j |\beta_j|^q$ *for given values of* q.

Yanjun Qi / UVA CS 4501-01-6501-07

In the example: Hybrid of Ridge and Lasso

Elastic Net regularization

$$\hat{\beta} \quad = \quad \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1$$

- The ℓ_1 part of the penalty generates a sparse model.
- The quadratic part of the penalty
  - Removes the limitation on the number of selected variables;
  - Encourages *grouping effect*;
  - Stabilizes the ℓ_1 regularization path.

9/9/14 32

Lasso Estimator

Elements of Statistical Learning, by Hastie, Tibshirani and Friedman

Ridge Regression

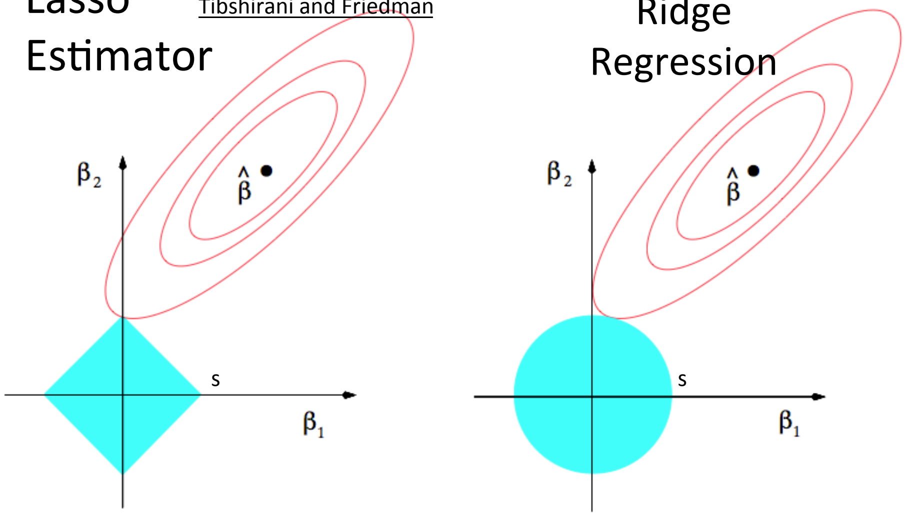
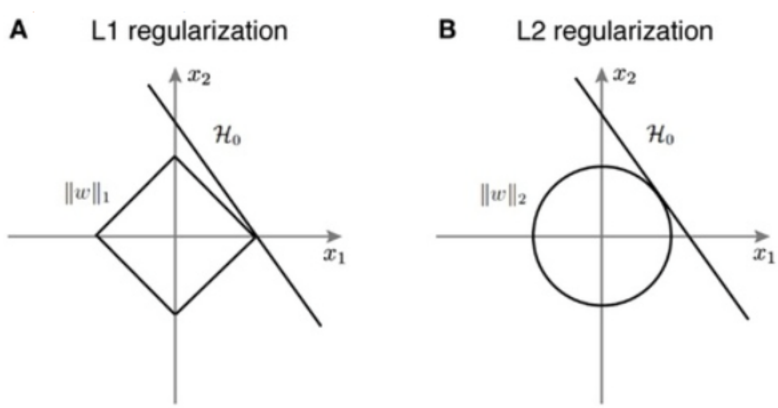


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*



due to the nature of L_1 norm, the viable solutions are limited to the corners, which are on one 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

Summary:
Regularized multivariate linear regression

- Model: $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$
  - LR estimation: $\min SSE = \sum \left(Y - \hat{Y}\right)^2$
  - LASSO estimation: $\min SSE = \sum_{i=1}^{n} \left(Y - \hat{Y}\right)^2 + \sum_{j=1}^{p} |\beta_j|$
  - Ridge regression estimation: $\min SSE = \sum_{i=1}^{n} \left(Y - \hat{Y}\right)^2 + \sum_{j=1}^{p} \beta_j^2$

35/54

Error on data + Regularization

Yanjun Qi / UVA CS 4501-01-6501-07

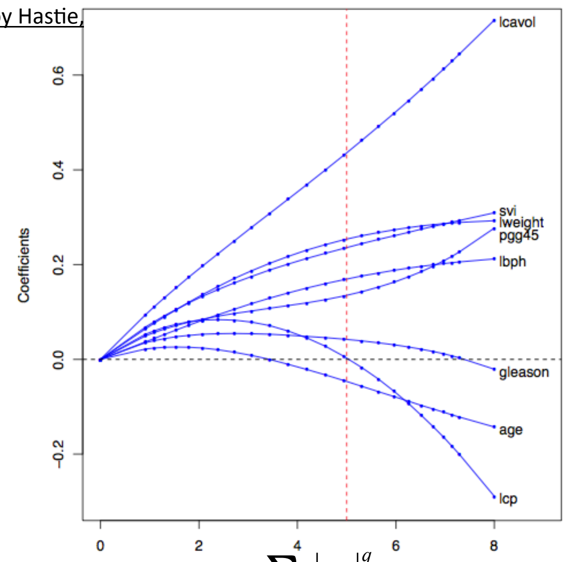
Extra

- Not required, though roughly covered during class
  - Subgradient
  - Coordinate descent based learning for Lasso

9/9/14 36

Elements of Statistical Learning, by Hastie, Tibshirani and Friedman

Ridge Regression

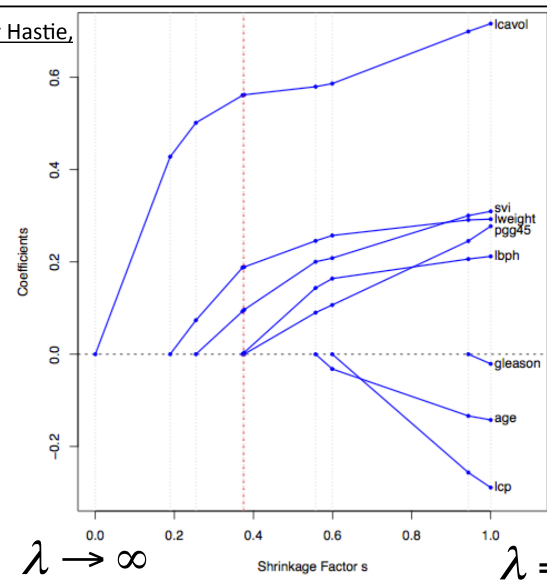


$$\sum_j |\beta_j|^q$$

$\lambda \to \infty$ $\lambda = 0$

Elements of Statistical Learning, by Hastie, Tibshirani and Friedman

Lasso Estimator



$\lambda \to \infty$ $\lambda = 0$

FIGURE 3.10. *Profiles of lasso coefficients, as the tuning parameter t is varied. Coefficients are plotted versus $s = t/\sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for ridge do not. The profiles are piece-wise linear, and so are computed only at the points displayed; see Section 3.4.4 for details.*

9/9/14

Today's Recap

- ❑ A bit more about Linear Regression Extension
  - ❑ Linear regression with predefined RBF basis
  - ❑ Locally weighted regression
- ❑ An Exemplar Application of Regression
  - ❑ Text based movie open weekend revenue prediction
- ❑ Linear Regression Models with Regularization
  - ❑ Ridge Regression
  - ❑ Lasso

References

- Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides

❑ Elements of Statistical Learning, by Hastie, Tibshirani and Friedman (page 61-69)