Introduction to Big Data Analysis
Regression

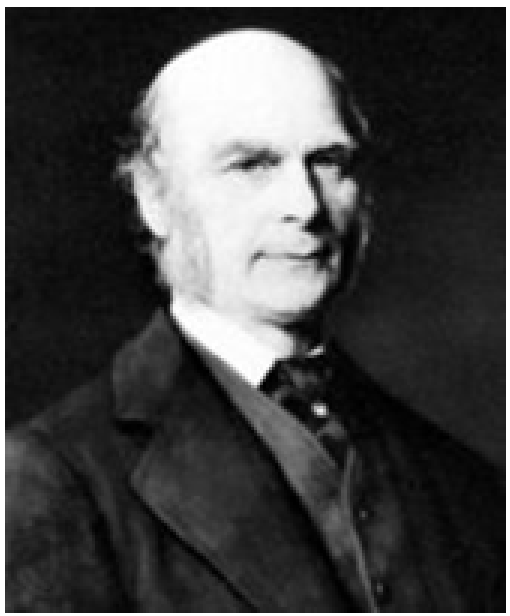
Zhen Zhang

Southern University of Science and Technology

Outlines

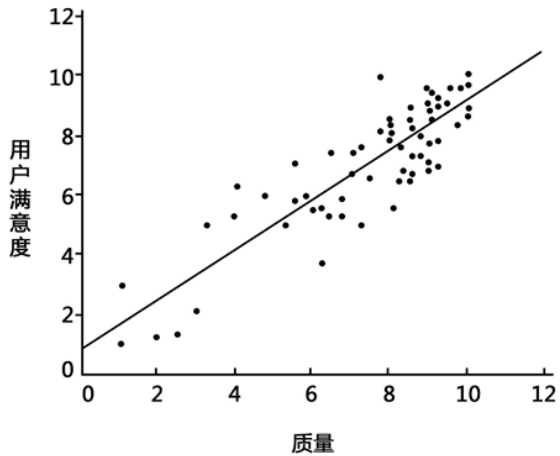
Regression

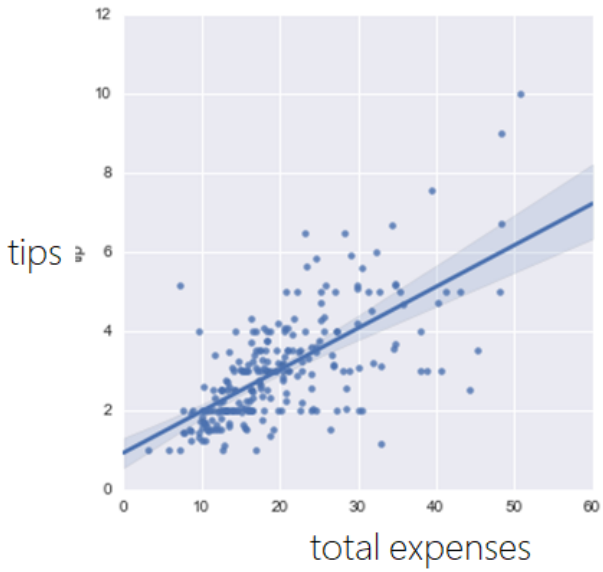
- Proposed by Francis Galton (left) and Karl Pearson (right), in the publication "Regression towards mediocrity in hereditary "
- The characteristics (e.g., height) in the offspring regress towards a mediocre point (mean) of that of their parents
- Generalization : predict the dependent variables y from the independent variables $\mathbf{x}$: $y = f(\mathbf{x})$ or $y = E[y|\mathbf{x}]$



Applications

- Predict medical expenses from the individual profiles of the patients
- Predict the scores on Douban from the quality of the movies
- Predict the tips from the total expenses



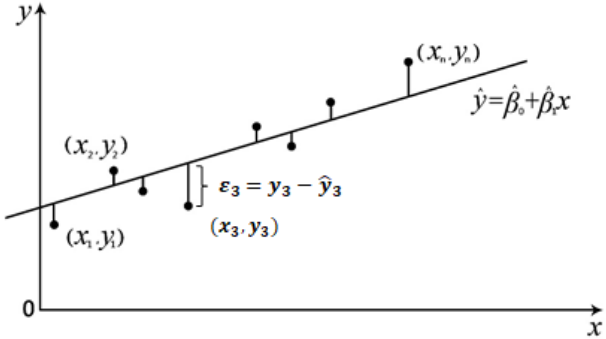


Outlines

Univariate Linear Model

- Linear model : $y = w_0 + w_1 x + \epsilon$, where w_0 and w_1 are regression coefficients, ϵ is the error or noise
- Assume $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where σ^2 is a fixed but unknown variance ; then $y|x \sim \mathcal{N}(w_0 + w_1 x, \sigma^2)$
- Assume the samples $\{x_i, y_i\}_{i=1}^n$ are generated from this conditional distribution, i.e., $y_i|x_i \sim \mathcal{N}(w_0 + w_1 x_i, \sigma^2)$
- Intuitively, find the best straight line (w_0 and w_1) such that the sample points fit it well, i.e., the residuals are minimized,

$$(\hat{w}_0, \hat{w}_1) = \arg\min_{w_0, w_1} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2$$

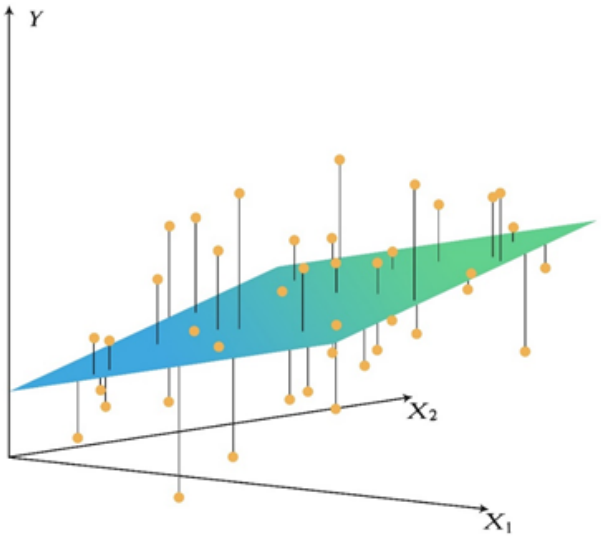


Multivariate Linear Model

- Linear model : $y = f(\mathbf{x}) + \epsilon = w_0 + w_1 x_1 + \cdots + w_p x_p + \epsilon$, where $w_0, w_1, \ldots, w_p$ are regression coefficients, $\mathbf{x} = (x_1, \ldots, x_p)^T$ is the input vector whose components are independent variables or attribute values, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the noise
- For the size n samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, let $\mathbf{y} = (y_1, \ldots, y_n)^T$ be the response or dependent variables, $\mathbf{w} = (w_0, w_1, \ldots, w_p)^T$, $\mathbf{X} = [\mathbf{1}_n, (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T] \in \mathbb{R}^{n \times (p+1)}$, and $\boldsymbol{\varepsilon} = (\epsilon_1, \ldots, \epsilon_n)^T \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$.

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon}$$

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

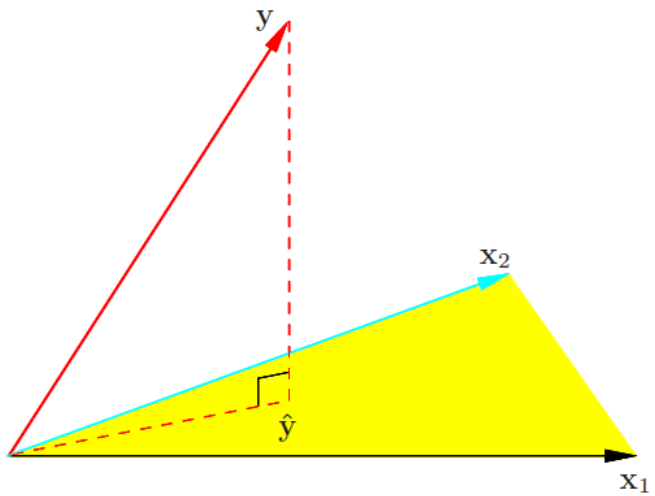


Least Square (LS)

- Minimize the total residual sum-of-squares :
  $RSS(\mathbf{w}) = \sum_{i=1}^{n}(y_i - w_0 - w_1 x_1 - \cdots - w_p x_p)^2 = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$
- When $\mathbf{X}^T\mathbf{X}$ is invertible, the minimizer $\hat{\mathbf{w}}$ satisfies

$$\nabla_{\mathbf{w}} RSS(\hat{\mathbf{w}}) = 0 \quad \Rightarrow \quad \hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- The prediction $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{P}\mathbf{y}$ is a projection of $\mathbf{y}$ onto the linear space spanned by the column vectors of $\mathbf{X}$; $\mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the projection matrix satisfying $\mathbf{P}^2 = \mathbf{P}$



Maximal Likelihood Estimate (MLE)

- A probabilistic viewpoint :
  $y|\mathbf{x} \sim \mathcal{N}(w_0 + w_1 x_1 + \cdots + w_p x_p, \sigma^2)$
- Likelihood function :
  $L(\mathbf{w}; \mathbf{X}, \mathbf{y}) = \mathrm{P}(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \Pi_{i=1}^{n} \mathrm{P}(y_i|\mathbf{x}_i, \mathbf{w})$ with
  $\mathrm{P}(y_i|\mathbf{x}_i, \mathbf{w}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i - w_0 - w_1 x_{i1} - \cdots - w_p x_{ip})^2}{2\sigma^2}}$
- Maximal likelihood estimate : given the samples from some unknown parametric distribution, find the parameters such that the samples the most probably seem to be drawn from that distribution, i.e., $\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} L(\mathbf{w}; \mathbf{X}, \mathbf{y})$
- Equivalent to maximize the log-likelihood function
  $l(\mathbf{w}; \mathbf{X}, \mathbf{y}) = \log L(\mathbf{w}; \mathbf{X}, \mathbf{y}) =$
  $-n \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_{i1} - \cdots - w_p x_{ip})^2$
- The same minimizer as LS : $\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

Projection by Orthogonalization

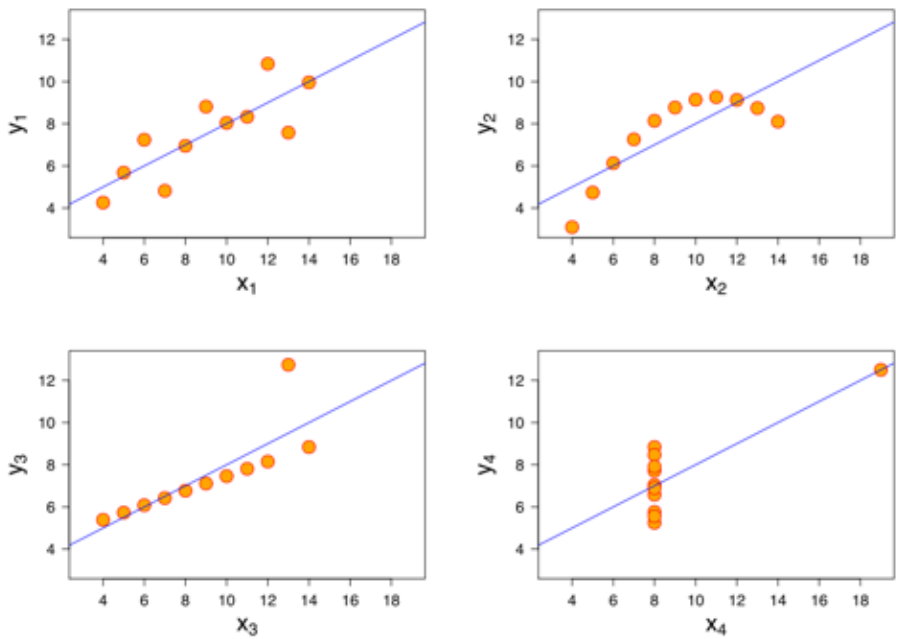
- Another useful formulation : let $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$, then OLS can be formulated by using the centralized data $\{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^{n} = \{\mathbf{x}_i - \bar{\mathbf{x}}, y_i - \bar{y}\}_{i=1}^{n}$, $RSS(\tilde{\mathbf{w}}) = \sum_{i=1}^{n}(\tilde{y}_i - w_1\tilde{x}_{i1} - \cdots - w_p\tilde{x}_{ip})^2 = \|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\|_2^2$, with $\hat{w}_0 = \bar{y} - \hat{\tilde{\mathbf{w}}}^T\bar{\mathbf{x}}$
- Ordinary least square (OLS) prediction $\hat{\mathbf{y}} = \mathbf{P}\mathbf{y}$ is the projection of $\mathbf{y}$ on the linear space spanned by the columns of $\mathbf{X}$, i.e., $\mathcal{X} = \mathrm{Span}\{\mathbf{x}_{\cdot,0}, \mathbf{x}_{\cdot,1}, \ldots, \mathbf{x}_{\cdot,p}\}$, recall that $\mathbf{x}_{\cdot,0} = \mathbf{1}_n$
- If $\{\mathbf{x}_{\cdot,0}, \mathbf{x}_{\cdot,1}, \ldots, \mathbf{x}_{\cdot,p}\}$ forms a set of orthonormal basis, then $\hat{\mathbf{y}} = \sum_{i=0}^{p} < \mathbf{y}, \mathbf{x}_{\cdot,i} > \mathbf{x}_{\cdot,i}$
- If not, we can first do orthogonalization by Gram-Schmidt procedure for the set $\{\mathbf{x}_{\cdot,0}, \mathbf{x}_{\cdot,1}, \ldots, \mathbf{x}_{\cdot,p}\}$
- Similar orthogonalization procedures can be done by QR decomposition or SVD of the matrix $\mathbf{X}^T\mathbf{X}$ (classic topics in numerical linear algebra)

Regression by Successive Orthogonalization

- The expansion of $\mathbf{y}$ on the standard orthonormal basis after Gram-Schmidt procedure can be summarised in the following algorithm :
  1. Initialize $\mathbf{z}_0 = \mathbf{x}_0 = \mathbf{1}_n$
  2. For $j = 1, \ldots, p$:
     Regress $\mathbf{x}_j$ on $\{\mathbf{z}_0, \ldots, \mathbf{z}_{j-1}\}$ to produce coefficients $\hat{\gamma}_{lj} = < \mathbf{z}_l, \mathbf{x}_j > / < \mathbf{z}_l, \mathbf{z}_l >$ with $l = 0, \ldots, j-1$ and residual vectors $\mathbf{z}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} \mathbf{z}_k$
  3. Regress $\mathbf{y}$ on the residual $\mathbf{z}_p$ to give the estimate $\hat{w}_p$
- If $\mathbf{x}_p$ is highly correlated with some of the other $\mathbf{x}_k$'s, the residual vector $\mathbf{z}_p$ will be close to zero ; in such situation, the coefficient $\hat{w}_p$ with small Z-score $\frac{\hat{w}_p}{\hat{\sigma}_p}$ could be thrown out, where $\hat{\sigma}_p^2 = \frac{\hat{\sigma}^2}{\|\mathbf{z}_p\|_2^2}$ is an estimate of $\mathrm{Var}(\hat{w}_p) = \frac{\sigma^2}{\|\mathbf{z}_p\|_2^2}$

Shortcomings of Fitting Nonlinear Data

- Evaluating the model by Coefficient of Determination R^2 :
  $R^2 := 1 - \frac{SS_{res}}{SS_{tot}} (= \frac{SS_{reg}}{SS_{tot}}$ for linear regression), where
  $SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2$ is the total sum of squares,
  $SS_{reg} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$ is the regression sum of squares, and
  $SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ is the residual sum of squares.
- The larger the R^2, the better the model



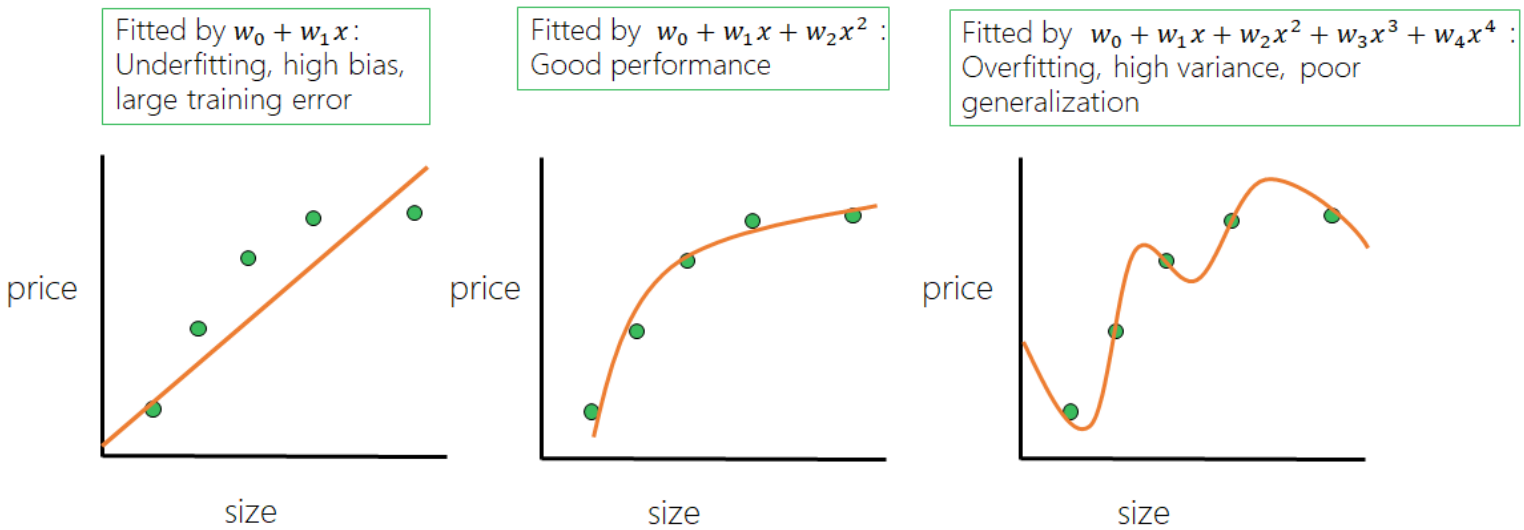
Multicolinearity

- If the columns of $\mathbf{X}$ are almost linearly dependent, i.e., multicolinearity, then $\det(\mathbf{X}^T\mathbf{X}) \approx 0$, the diagonal entries in $(\mathbf{X}^T\mathbf{X})^{-1}$ is quite large. This implies the variances of $\hat{\mathbf{w}}$ get large, and the estimate is not accurate
- Eg : 10 samples are drawn from the true model $y = 10 + 2x_1 + 3x_2 + \epsilon$; the LS estimator is $\hat{w}_0 = 11.292$, $\hat{w}_1 = 11.307$, $\hat{w}_2 = -6.591$, far from the true coefficients; correlation coefficient is $r_{12} = 0.986$
- Remedies : ridge regression, principal component regression, partial least squares regression, etc.

No.	1	2	3	4	5	6	7	8	9	10
x_1	1.1	1.4	1.7	1.7	1.8	1.8	1.9	2.0	2.3	2.4
x_2	1.1	1.5	1.8	1.7	1.9	1.8	1.8	2.1	2.4	2.5
ε_i	0.8	-0.5	0.4	-0.5	0.2	1.9	1.9	0.6	-1.5	-1.5
y_i	16.3	16.8	19.2	18.0	19.5	20.9	21.1	20.9	20.3	22.0

Overfitting

- Easily to be overfitted when introducing more variables, e.g., regress housing price with housing size
- The high degree model also fits the noises in the training data, so generalizes poorly to new data
- Remedy : regularization

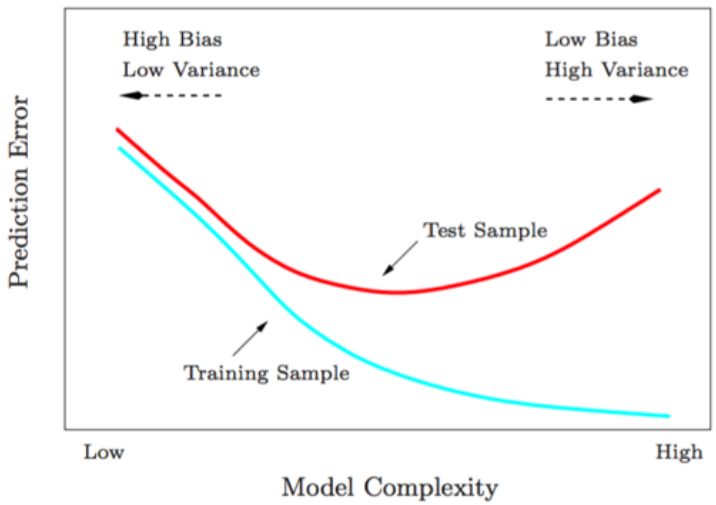


Bias-Variance Decomposition

- Bias-variance decomposition of generalization error in L^2 loss :

$$\mathrm{E}_{train} R_{exp}(\hat{f}(\mathbf{x})) = \mathrm{E}_{train}\mathrm{E}_P[(y-\hat{f}(\mathbf{x}))^2|\mathbf{x}] = \underbrace{\mathrm{Var}(\hat{f}(\mathbf{x}))}_{\text{variance}} + \underbrace{\mathrm{Bias}^2(\hat{f}(\mathbf{x}))}_{\text{bias}} + \underbrace{\sigma^2}_{\text{noise}}$$

  where $P = P(y|\mathbf{x})$ is the conditional probability of y given $\mathbf{x}$
- Bias : $\mathrm{Bias}(\hat{f}(\mathbf{x})) = \mathrm{E}_{train}\hat{f}(\mathbf{x}) - f(\mathbf{x})$ is the average accuracy of prediction for the model (deviation from the truth)
- Variance : $\mathrm{Var}(\hat{f}(\mathbf{x})) = \mathrm{E}_{train}(\hat{f}(\mathbf{x}) - \mathrm{E}_{train}\hat{f}(\mathbf{x}))^2$ is the variability of the model prediction due to different data set (stability)



Bias-Variance Decomposition (Derivation)

Model $y = f(\mathbf{x}) + \epsilon$, with $\mathrm{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma^2$ (system error)

$$
\begin{aligned}
\mathrm{E}_{train} R_{exp}(\hat{f}(\mathbf{x})) =& \mathrm{E}_P[(y - f(\mathbf{x}))^2 | \mathbf{x}] + \mathrm{E}_{train}[(f(\mathbf{x}) - \hat{f}(\mathbf{x}))^2] \\
&+ 2 \underbrace{\mathrm{E}_{train} \mathrm{E}_P[(y - f(\mathbf{x}))(f(\mathbf{x}) - \hat{f}(\mathbf{x})) | \mathbf{x}]}_{\text{vanishes since } \mathrm{E}_P(y - f(\mathbf{x}) | \mathbf{x}) = 0} \\
=& \sigma^2 + \mathrm{E}_{train}[(f(\mathbf{x}) - \mathrm{E}_{train} \hat{f}(\mathbf{x}))^2] + \mathrm{E}_{train}[(\mathrm{E}_{train} \hat{f}(\mathbf{x}) - \hat{f}(\mathbf{x}))^2] \\
&+ 2 \underbrace{\mathrm{E}_{train}[(f(\mathbf{x}) - \mathrm{E}_{train} \hat{f}(\mathbf{x}))(\mathrm{E}_{train} \hat{f}(\mathbf{x}) - \hat{f}(\mathbf{x}))]}_{\text{vanishes since } \mathrm{E}_{train}[\mathrm{E}_{train} \hat{f}(\mathbf{x}) - \hat{f}(\mathbf{x})] = 0} \\
=& \sigma^2 + \mathrm{Bias}^2(\hat{f}(\mathbf{x})) + \mathrm{Var}(\hat{f}(\mathbf{x}))
\end{aligned}
$$

The more complicated the model, the lower the bias, but the higher the variance.

Bias-Variance Decomposition : kNN Regression

- kNN can be used to do regression if the mode (majority vote) is replaced by mean : $\hat{f}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_{(i)} \in N_k(\mathbf{x})} y_{(i)}$
- Generalization error of kNN regression is

$$\begin{aligned} \mathrm{E}_{train} R_{exp}(\hat{f}(\mathbf{x})) =& \sigma^2 + (f(\mathbf{x}) - \frac{1}{k} \sum_{\mathbf{x}_{(i)} \in N_k(\mathbf{x})} f(\mathbf{x}_{(i)}))^2 \\ & + \underbrace{\mathrm{E}_{train} \left[\frac{1}{k} \sum_{\mathbf{x}_{(i)} \in N_k(\mathbf{x})} (y_{(i)} - f(\mathbf{x}_{(i)})) \right]^2}_{\frac{1}{k}\sigma^2} \end{aligned}$$

  where we have used the fact that $\mathrm{E}_{train} y_i = f(\mathbf{x}_i)$ and $\mathrm{Var}(y_i) = \sigma^2$.
- For small k, overfitting, bias $\searrow$, variance $\nearrow$
- For large k, underfitting, bias $\nearrow$, variance $\searrow$

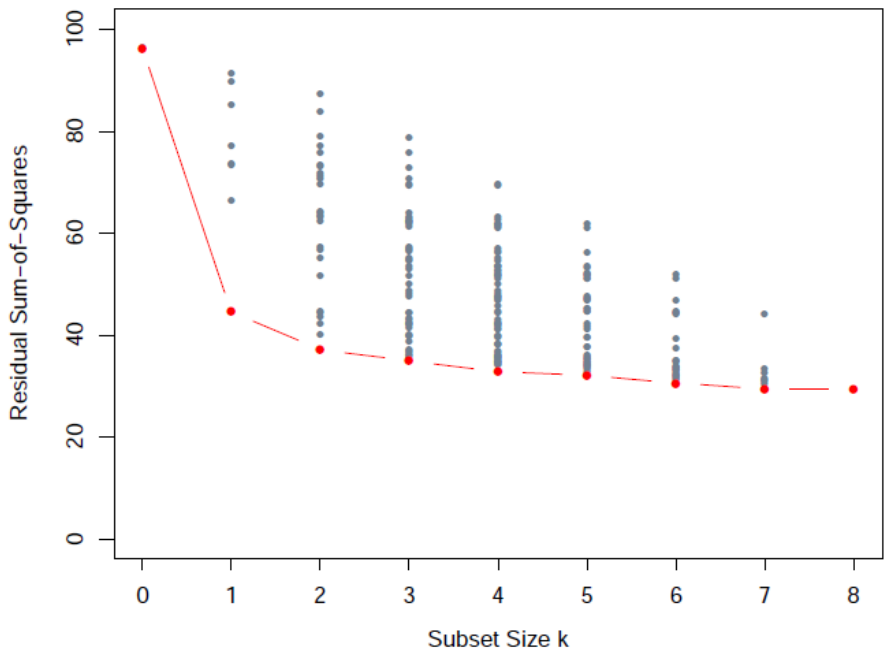
Outlines

Regularization by Subset Selection

- In high dimensions, the more the input attributes, the larger the variance
- Shrinking some coefficients or setting them to zero can reduce the overfitting
- Using less input variables also help interpretation with the most important variables
- Subset selection： retaining only a subset of the variables, while eliminating the rest variables from the model
- Best-subset selection : find for each $k \in \{0, 1, \ldots, p\}$ the subset $S_k \subset \{1, \ldots, p\}$ of size k that gives the smallest $RSS(\mathbf{w}) = \sum_{i=1}^{n}(y_i - w_0 - \sum_{j \in S_k} w_j x_{ij})^2$

Best-Subset Selection

- The best subset of size $k+1$ may not include the the variables in the best subset of size k
- The RSS of the best subset of size k is not necessarily decreasing with k
- Choose k based on bias-variance tradeoff, usually by AIC and BIC, or practically by cross-validation

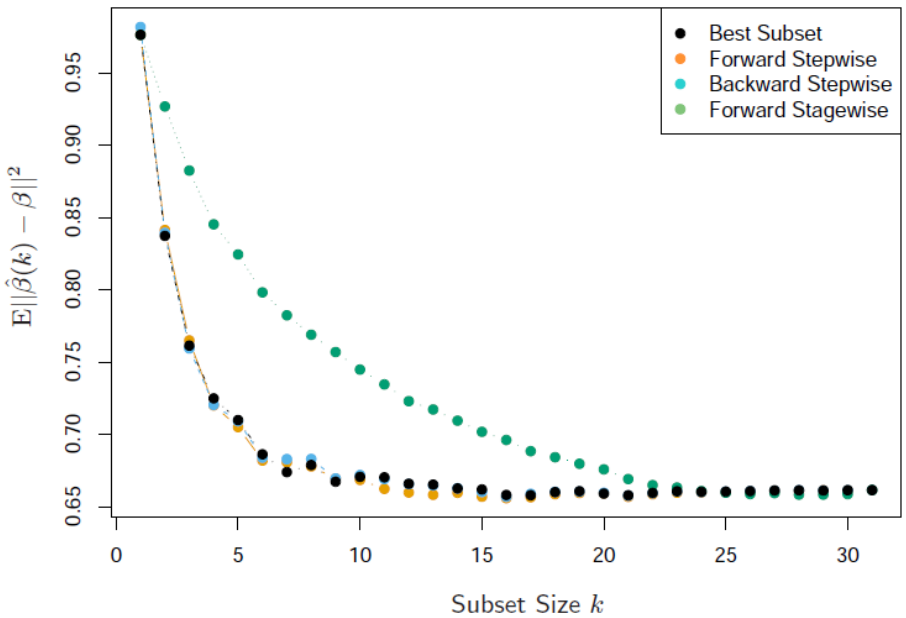


Forward (Backward) Stepwise Selection

- Forward-stepwise selection : start with the intercept $\bar{y}$, then sequentially add into the model the variables that improve the fit most (reduce RSS most)
- QR factorization helps search the candidate variables to add
- Greedy algorithm : the solution could be sub-optimal
- Computationally more efficient than best-subset selection ; statistically the constrained search enjoys lower variance than best-subset selection
- Backward-stepwise selection : start with the full model, then sequentially delete from the model the variables that has the least impact on the fit most (the candidate for dropping is the variable with the smallest Z-score) ; can only be used when $n > p$ in order to fit the full model by OLS

Forward-Stagewise (FS) Selection

- Starts with the intercept and centered variables with 0 coefficients
- At each step, identify the variables (among all variables) most correlated with the current residual, then regress the residual on this chosen variable and increment the current coefficient with the new regression coefficient
- Ends when no variables are correlated with the residual (arrive at the OLS fit when $n > p$)
- Slower than forward-stepwise : the other variables and their coefficients are not changed at each step except the chosen variable

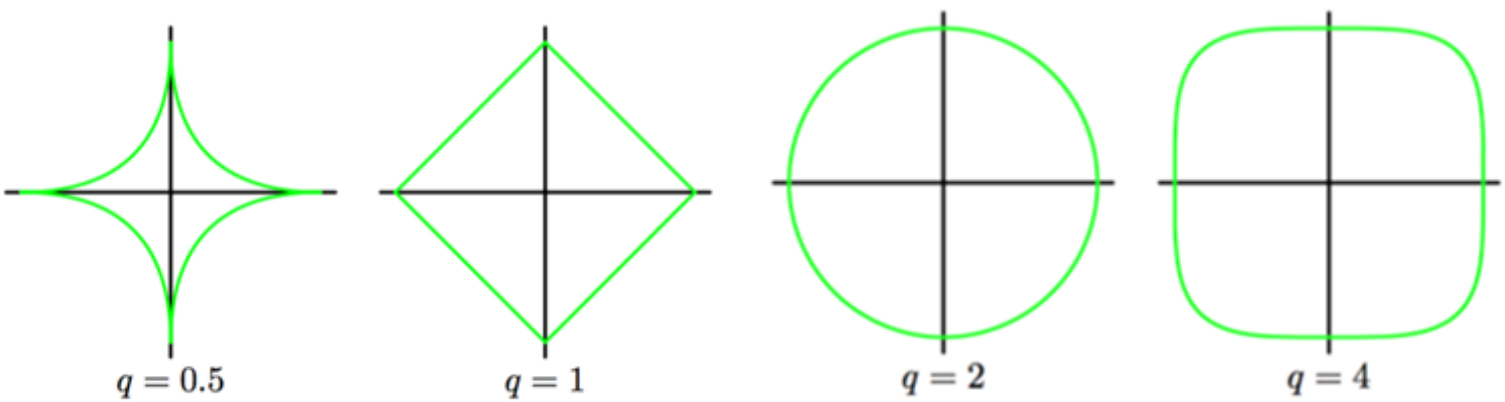


Regularization by Penalties

- Add a penalty term, in general l_q-norm

$$\sum_{i=1}^{n}(y_i - w_0 - w_1 x_1 - \cdots - w_p x_p)^2 + \lambda \|\mathbf{w}\|_q^q$$
$$= \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_q^q$$

- $q = 2$: ridge regression
- $q = 1$: LASSO regression



Ridge Regression

- The optimization problem turns to be

$$\begin{aligned}\hat{\mathbf{w}} &= \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_1 - \cdots - w_p x_p)^2 + \lambda \|\mathbf{w}\|_2^2 \\ &= \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2\end{aligned}$$

- $\lambda \geqslant 0$ is a fixed parameter which has to be tuned by cross-validation
- Equivalent to the constraint minimization problem :

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2, \qquad \text{subject to} \qquad \|\mathbf{w}\|_2 \leqslant \mu,$$

  where $\mu \geqslant 0$ is a prescribed threshold (tuning parameter)
- The large λ corresponds to the small μ.

Solving Ridge Regression

- Easy to show that $\hat{\mathbf{w}}^{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{p+1})^{-1}\mathbf{X}^T\mathbf{y}$
- The estimator is also a projection of $\mathbf{y}$: $\hat{\mathbf{y}}^{ridge} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{p+1})^{-1}\mathbf{X}^T\mathbf{y}$
- $\mathbf{X}$ can be diagonalized by SVD : $\mathbf{X} = \mathbf{PDQ}$ with $\mathbf{D} = \mathrm{diag}(\nu_1, \ldots, \nu_{p+1})$, and $\mathbf{P} \in \mathbb{R}^{n\times(p+1)}$, $\mathbf{Q} \in \mathbb{R}^{(p+1)\times(p+1)}$ being orthogonal matrices ($\mathbf{P}^T\mathbf{P} = \mathbf{I}_{p+1}$)
- $\hat{\mathbf{y}}^{ridge} = \mathbf{P}\mathrm{diag}(\frac{\nu_1^2}{\nu_1^2+\lambda}, \ldots, \frac{\nu_{p+1}^2}{\nu_{p+1}^2+\lambda})\mathbf{P}^T\mathbf{y}$, while $\hat{\mathbf{y}}^{OLS} = \mathbf{PP}^T\mathbf{y}$
- In the spectral space, the ridge regression estimator is a shrinkage of the OLS estimator ($\lambda = 0$)

Bayesian Viewpoint of Ridge Regression

- Given $\mathbf{X}$ and $\mathbf{w}$, the conditional distribution of $\mathbf{y}$ is $\mathrm{P}(\mathbf{y}|\mathbf{X},\mathbf{w}) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}) \propto \exp\big(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})\big)$
- In addition, assume $\mathbf{w}$ has a prior distribution $\mathrm{P}(\mathbf{w}) = \mathcal{N}(\mu_0, \mathbf{\Lambda}_0) \propto \exp\big(-\frac{1}{2}(\mathbf{w}-\mu_0)^T\Lambda_0^{-1}(\mathbf{w}-\mu_0)\big)$
- By Bayes theorem, the posterior distribution of $\mathbf{w}$ given the data $\mathbf{X}$ and $\mathbf{y}$ is

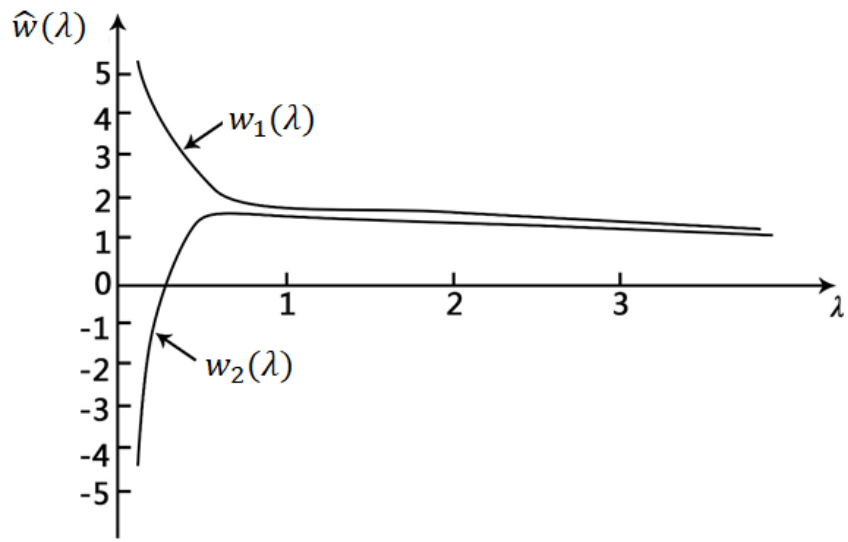
$$\begin{aligned}\mathrm{P}(\mathbf{w}|\mathbf{X},\mathbf{y}) &\propto \mathrm{P}(\mathbf{y}|\mathbf{X},\mathbf{w})\mathrm{P}(\mathbf{w}) \\ &\propto \exp\Big(-\frac{1}{2\sigma^2}(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{y}^T\mathbf{X}\mathbf{w}) \\ &\qquad -\frac{1}{2}(\mathbf{w}^T\mathbf{\Lambda}_0^{-1}\mathbf{w} - 2\mu_0^T\mathbf{\Lambda}_0^{-1}\mathbf{w})\Big) \\ &\propto \exp\Big(-\frac{1}{2}(\mathbf{w}-\mu_m)^T\Lambda_m^{-1}(\mathbf{w}-\mu_m)\Big)\end{aligned}$$

  where $\Lambda_m = (\frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X} + \mathbf{\Lambda}_0^{-1})^{-1}$ and $\mu_m = \mathbf{\Lambda}_m(\frac{1}{\sigma^2}\mathbf{X}^T\mathbf{y} + \mathbf{\Lambda}_0^{-1}\mu_0)$
- If $\mu_0 = 0$ and $\mathbf{\Lambda}_0 = \frac{\sigma^2}{\lambda}\mathbf{I}_{p+1}$, then $\hat{\mathbf{w}} = \mu_m = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{p+1})^{-1}\mathbf{X}^T\mathbf{y}$ maximizes the posterior probability $\mathrm{P}(\mathbf{w}|\mathbf{X},\mathbf{y})$

Ridge Trace

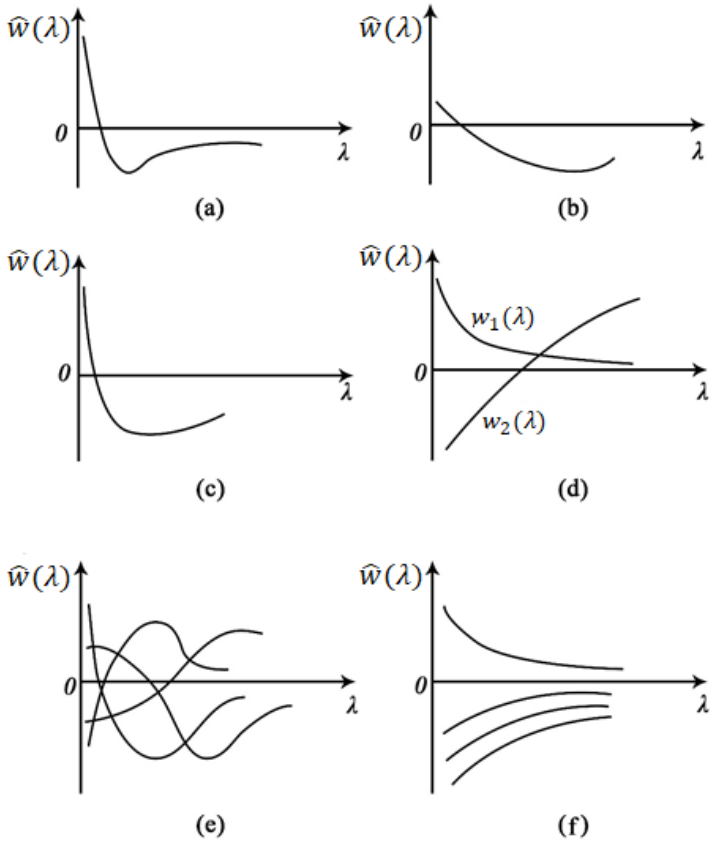
- The functional plot of $\hat{\mathbf{w}}^{ridge}(\lambda)$ with λ is called ridge trace
- The large variations in ridge trace indicate the multicolinearity in variables
- When $\lambda \in (0, 0.5)$, the ridge traces have large variations, it suggests to choose $\lambda = 1$

λ	0	0.1	0.15	0.2	0.3	0.4	0.5	1.0	1.5	2.0	3.0
$\hat{w}_1^{ridge}(\lambda)$	11.31	3.48	2.99	2.71	2.39	2.20	2.06	1.66	1.43	1.27	1.03
$\hat{w}_2^{ridge}(\lambda)$	-6.59	0.63	1.02	1.21	1.39	1.46	1.49	1.41	1.28	1.17	0.98



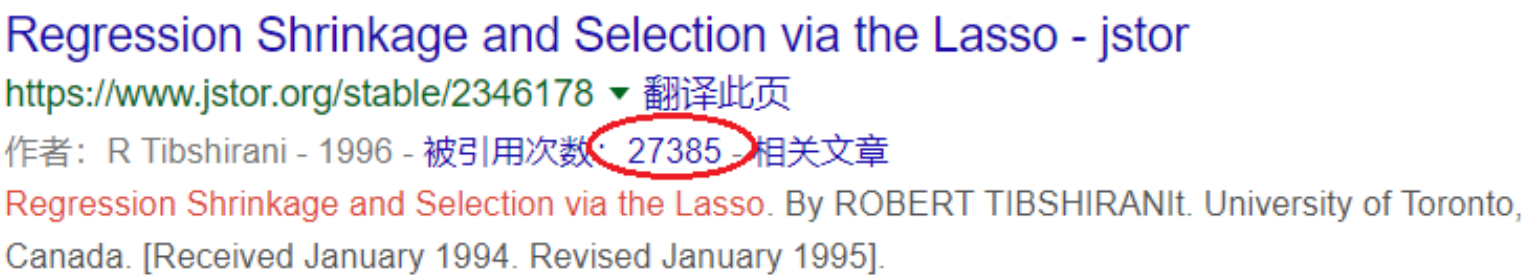
Reading from Ridge Trace

- Before plot ridge trace, do scaling for the variables
- The coefficients with stable trace and small absolute values should have little influence on y, as in (a)
- The coefficients with large stable absolute values should have great impact on y, as in (b) and (c)
- The ridge traces of the coefficients of two variables are not stable, but the sum of the coefficients is stable. This implies the multicolinearity as in (d)
- The stable ridge traces of all variables suggest good performance using OLS as in (f)



LASSO Regression

- Proposed by R. Tibshirani, short for "Least Absolute **Shrinkage and Selection** Operator"
- Can be used to estimate the coefficients and select the important variables simultaneously
- Reduce the model complexity, avoid overfitting, and improve the generalization ability
- Also improve the model interpretability



LASSO Formulation

- The optimization problem

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} E(\mathbf{w})$$

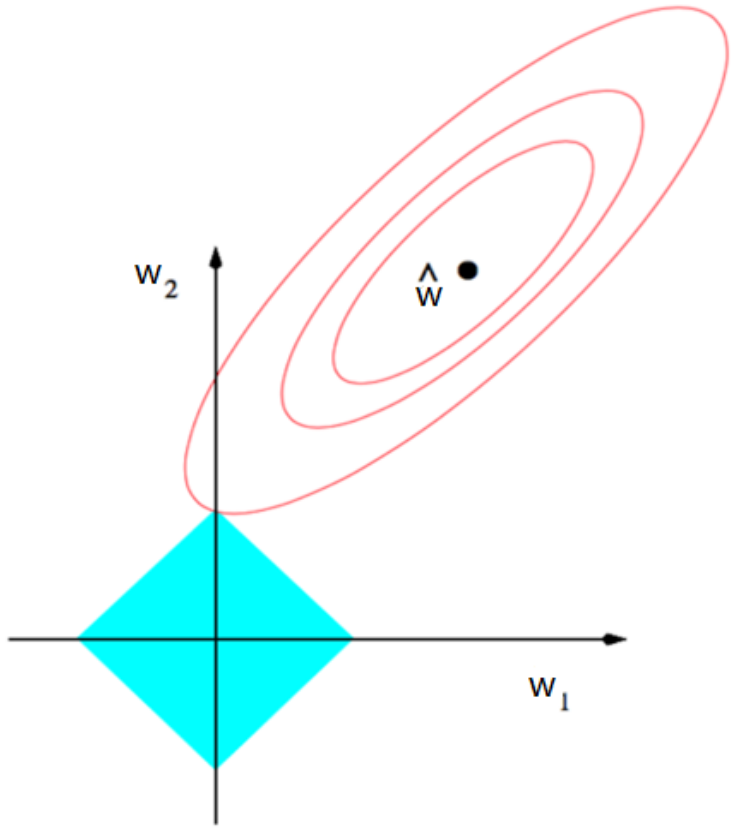
$$E(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_1$$

- Equivalent to the constraint minimization problem :

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2,$$

$$\text{subject to} \qquad \|\mathbf{w}\|_1 \leqslant \mu,$$

- The large λ corresponds to the small μ.
- The optimal solution is sparse with $\hat{w}_2 = 0$

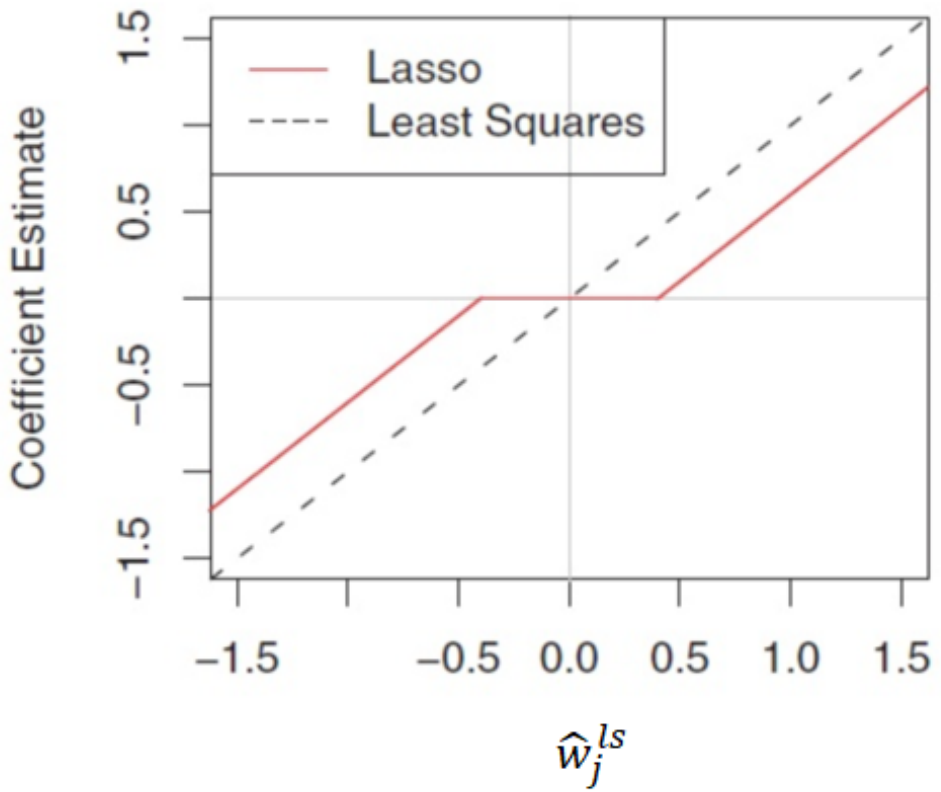


Solving LASSO Regression

- Assume $\mathbf{X}^T\mathbf{X} = \mathbf{I}_{p+1}$, then $\hat{\mathbf{w}}^{OLS} = \mathbf{X}^T\mathbf{y}$
- $\partial_{\mathbf{w}} E(\mathbf{w}) = \mathbf{w} - \mathbf{X}^T\mathbf{y} + \lambda(\partial|w_0| \times \cdots \times \partial|w_p|)$
- $\mathbf{0} \in \partial_{\mathbf{w}} E(\hat{\mathbf{w}}^{lasso})$ implies $0 \in \hat{w}_i^{lasso} - \hat{w}_i^{OLS} + \lambda\partial|\hat{w}_i^{lasso}|$
- If $\hat{w}_i^{lasso} > 0$, $\partial|\hat{w}_i^{lasso}| = \{1\}$, and $\hat{w}_i^{lasso} = \hat{w}_i^{OLS} - \lambda$ with $\hat{w}_i^{OLS} > \lambda$
- If $\hat{w}_i^{lasso} < 0$, $\partial|\hat{w}_i^{lasso}| = \{-1\}$, and $\hat{w}_i^{lasso} = \hat{w}_i^{OLS} + \lambda$ with $\hat{w}_i^{OLS} < -\lambda$
- If $\hat{w}_i^{lasso} = 0$, $\partial|\hat{w}_i^{lasso}| = [-1, 1]$, and $\hat{w}_i^{OLS} \in [-\lambda, \lambda]$
- In summary, $\hat{w}_i^{lasso} = (|\hat{w}_i^{OLS}| - \lambda)_+ \mathrm{sign}(\hat{w}_i^{OLS})$

Shrinkage and Selection Property of LASSO

$\hat{w}_i^{lasso} = (|\hat{w}_i^{OLS}| - \lambda)_+ \mathrm{sign}(\hat{w}_i^{OLS})$ is called soft thresholding of $\hat{w}_i^{OLS}$, where $(a)_+ = \max(a, 0)$ is the positive part of a



Maximum A Posteriori (MAP) Estimation

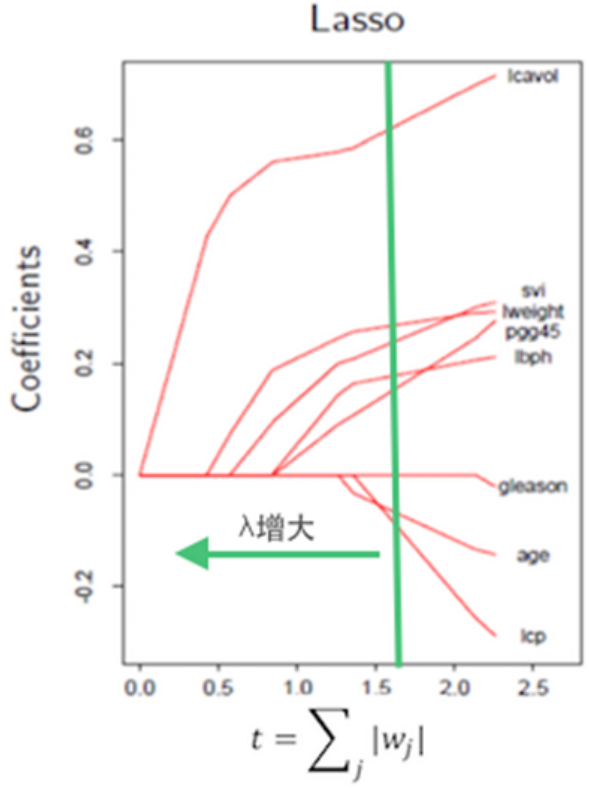
- Given θ, the conditional distribution of $\mathbf{y}$ is $\mathrm{P}(\mathbf{y}|\theta)$
- In addition, assume the parameter θ has a prior distribution $\mathrm{P}(\theta)$
- The posterior distribution of θ given the data $\mathbf{y}$ is $\mathrm{P}(\theta|\mathbf{y}) \propto \mathrm{P}(\mathbf{y}|\theta)\mathrm{P}(\theta)$
- MAP choose the point of maximal posterior probability :

$$\hat{\theta}^{MAP} = \arg\max_{\theta} \mathrm{P}(\theta|\mathbf{y}) = \arg\max_{\theta} \big(\log \mathrm{P}(\mathbf{y}|\theta) + \log \mathrm{P}(\theta) \big)$$

- If $\theta = \mathbf{w}$, and we choose the log-prior proportional to $\lambda\|\mathbf{w}\|_2^2$ (i.e., the normal prior $\mathcal{N}(0, \frac{\sigma^2}{\lambda}\mathbf{I})$), we recover the ridge regression
- If the log-prior is proportional to $\lambda\|\mathbf{w}\|_1$, i.e., the prior is the tensor product of Laplace (or double exponential) distribution Laplace$(0, \frac{2\sigma^2}{\lambda})$
- Different log-prior lead to different penalties (regularization), but this is not the case in general : some penalties may not be the logarithms of probability distributions, some other penalties depend on the data (prior is independent of the data)

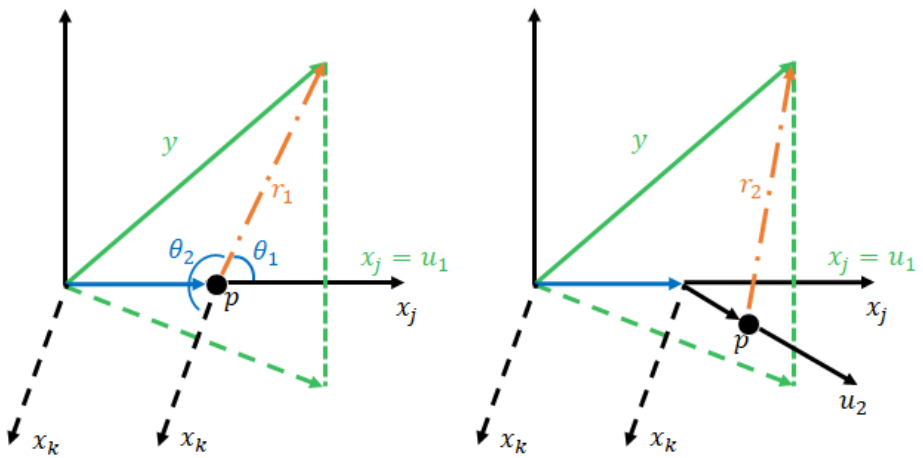
LASSO Path

- When λ varies, the values of the coefficients form paths (regularization paths)
- The paths are piecewise linear with the same change points, may cross the x-axis many times
- In practice, choose λ by cross-validation



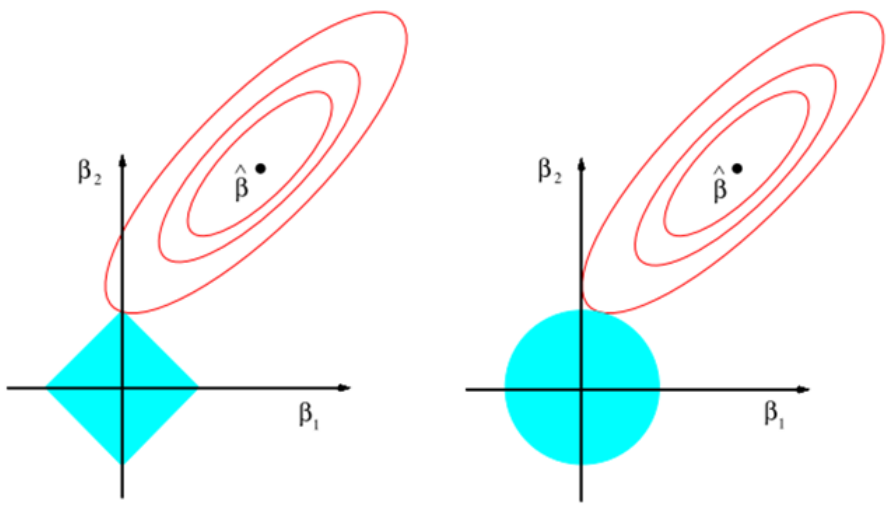
Solving LASSO by LARS (Hastie and Efron)

1. Start with all coefficients w_i equal to zero
2. Find the predictor x_i most correlated with y
3. Increase the coefficient w_i in the direction of the sign of its correlation with y. Take residuals $r = y - \hat{y}$ along the way. Stop when some other predictor x_k has as much correlation with r as x_i has
4. Increase (w_i, w_k) in their joint least squares direction, until some other predictor x_m has as much correlation with the residual r
5. Continue until all predictors are in the model



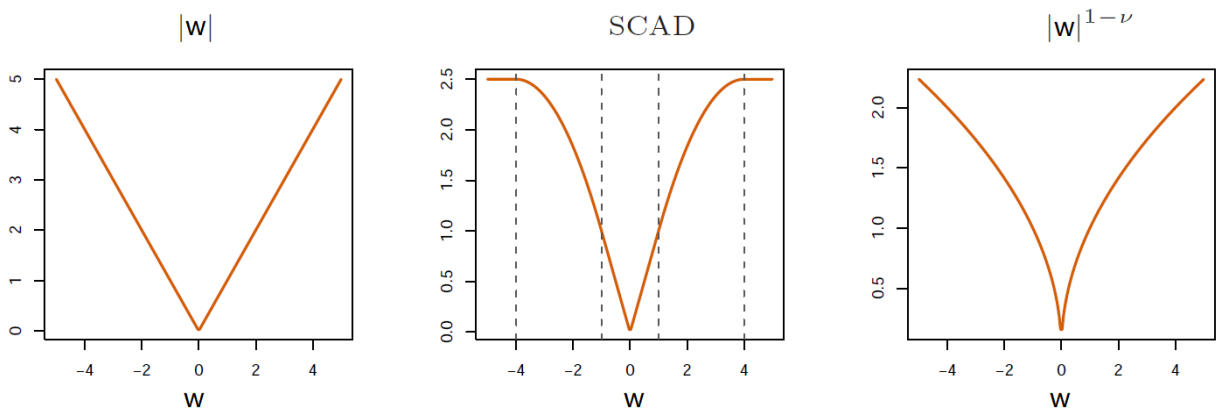
Other Solvers

- "glmnet" by Friedman, Hastie and Tibshirani, implemented by coordinate descent, can be used in linear regression, logistic regression, etc., with LASSO (l_1), ridge (l_2) and elastic net ($l_1 + l_2$) regularization terms
- Why LASSO seeks the sparse solution in comparison with ridge ?



Related Regularization Models

- Elastic net : $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}\|_2^2 + \lambda_2 \|\mathbf{w}\|_1$
- Group LASSO : $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \sum_{g=1}^{G} \lambda_g \|\mathbf{w}_g\|_2$, where $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_G)$ is the group partition of $\mathbf{w}$
- Dantzig Selector : $\min_{\mathbf{w}} \|\mathbf{w}\|_1$, subject to $\|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})\|_\infty \leqslant \mu$
- Smoothly clipped absolute deviation (SCAD) penalty by Fan and Li (2005) : replace the penalty $\lambda \sum_{i=0}^{p} |w_i|$ by $\sum_{i=0}^{p} J_a(w_i, \lambda)$, where $J_a(x, \lambda)$ satisfies (for $a \geqslant 2$) : $\frac{\mathrm{d}J_a}{\mathrm{d}x} = \lambda \mathrm{sign}(x)\left(I(|x| \leqslant \lambda) + \frac{(a\lambda - |x|)_+}{(a-1)\lambda} I(|x| > \lambda)\right)$
- Adaptive LASSO : weighted penalty $\sum_{i=0}^{p} \mu_i |w_i|$ where $\mu_i = \frac{1}{|\hat{w}_i^{OLS}|^\nu}$ with $\nu > 0$, as an approximation to $|w_i|^{1-\nu}$, non-convex penalty



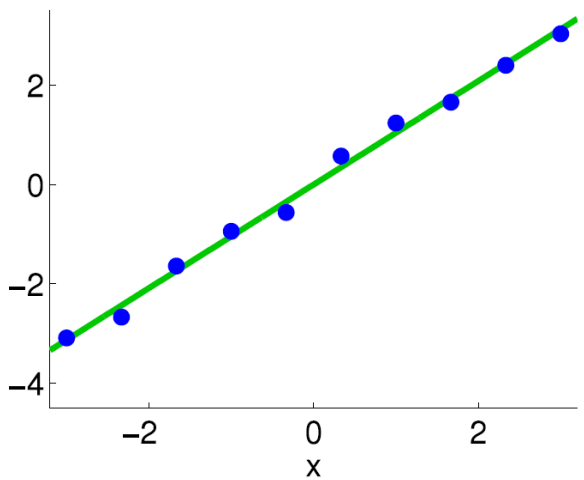
Outlines

Problem with Outliers - Robustness

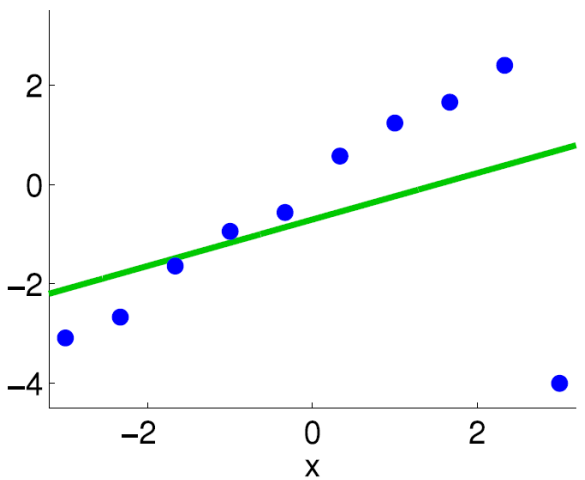
- l_2-loss minimization is sensitive to outliers (non-robust)
- It penalizes greatly for the large residual, probably at outliers
- Consider l_2-regression towards a constant model $f_\theta(x) = \theta$:

$$\hat{\theta}_{LS} = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta)^2 = \frac{1}{n} \sum_{i=1}^{n} y_i = Mean(\{y_i\})$$

- If l_1-loss is used : $\hat{\theta}_{LS} = \arg\min_{\theta} \sum_{i=1}^{n} |y_i - \theta| = Median(\{y_i\})$



(a) Without outliers



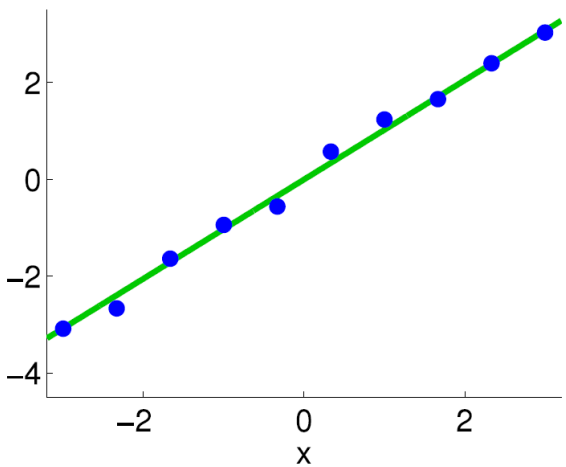
(b) With an outlier

Least Absolute Deviations Regression

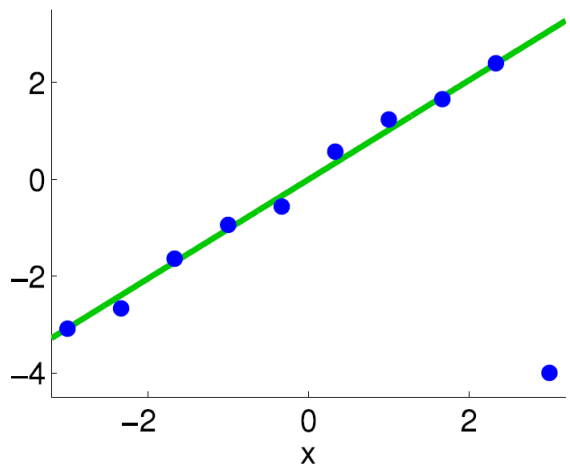
- With l_1-loss : $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_1$
- Recast to linear programming (Boyd & Vandenberghe, 2004) :

$$\min_{\mathbf{w}} \sum_{i=1}^{n} r_i$$

$$\text{subject to} \quad -r_i \leqslant y_i - \mathbf{w}^T \mathbf{x}_i \leqslant r_i, \quad i = 1, \ldots, n$$



(a) Without outliers



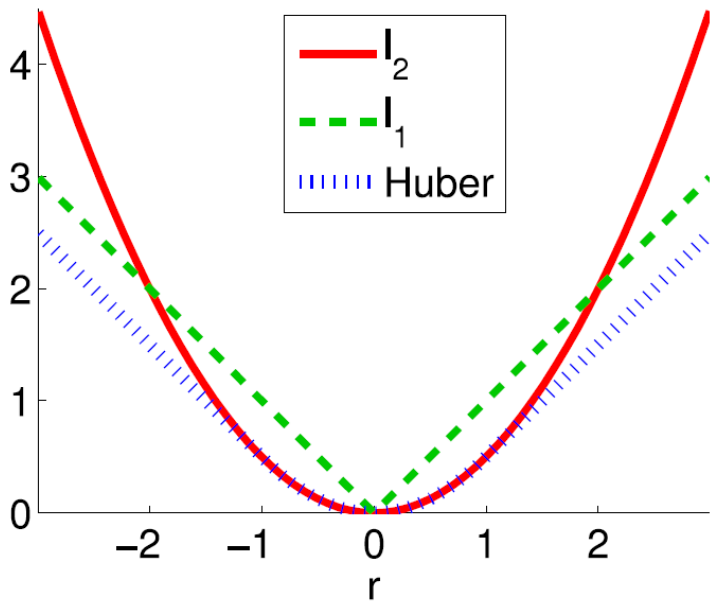
(b) With an outlier

Huber Loss Minimization

- Combine l_2-loss with l_1-loss : $L(y, f) = \rho_{Huber}(y - f)$, where

$$\rho_{Huber}(r) = \begin{cases} \frac{r^2}{2}, & |r| \leqslant \eta \\ \eta|r| - \frac{\eta^2}{2}, & |r| > \eta \end{cases}$$

- Huber loss minimization : $\min_{\mathbf{w}} J(\mathbf{w}) \stackrel{\Delta}{=} \sum_{i=1}^{n} \rho_{Huber}(y_i - \mathbf{w}^T \mathbf{x}_i)$



Optimizing Huberized Regression

- Gradient descent :

$$\rho'_{Huber}(r) = \begin{cases} r, & |r| \leqslant \eta \\ -\eta, & r < -\eta \\ \eta, & r > \eta \end{cases}$$

  then updating $\mathbf{w} \leftarrow \mathbf{w} + \tau \sum_{i=1}^{n} \mathbf{x}_i \rho'_{Huber}(y_i - \mathbf{w}^T \mathbf{x}_i)$ with τ being learning rate (or step size)
- Stochastic gradient descent : replace the summation by randomly selecting one sample $(\mathbf{x}_i, y_i)$ at each step
- Can be used in combination with l_1 regularization (like Lasso)

Iteratively Reweighted Least Square (IRLS)

- Quadratically upper bound the absolute-value part of Huber loss :

$$\eta|r| - \frac{\eta^2}{2} \leqslant \frac{\eta}{2c}r^2 + \frac{\eta c}{2} - \frac{\eta^2}{2}, \quad c > 0$$

  this upper bound touches Huber loss at $r = \pm c$ (i.e., "=" holds)

- Take $c_i^{(k)} = |y_i - (\mathbf{w}^{(k)})^T \mathbf{x}_i|$ to be the absolute residual for the i-th training sample at k-th step iteration, instead of directly minimize Huber loss, we can minimize its k-th upper bound :

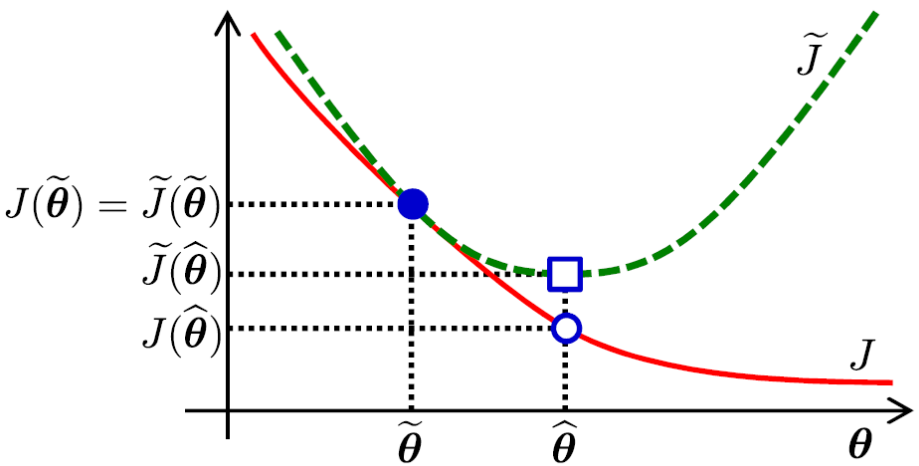
$$\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}} \tilde{J}(\mathbf{w}) \triangleq \frac{1}{2}\sum_{i=1}^{n} \alpha_i^{(k)}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 + C^{(k)},$$

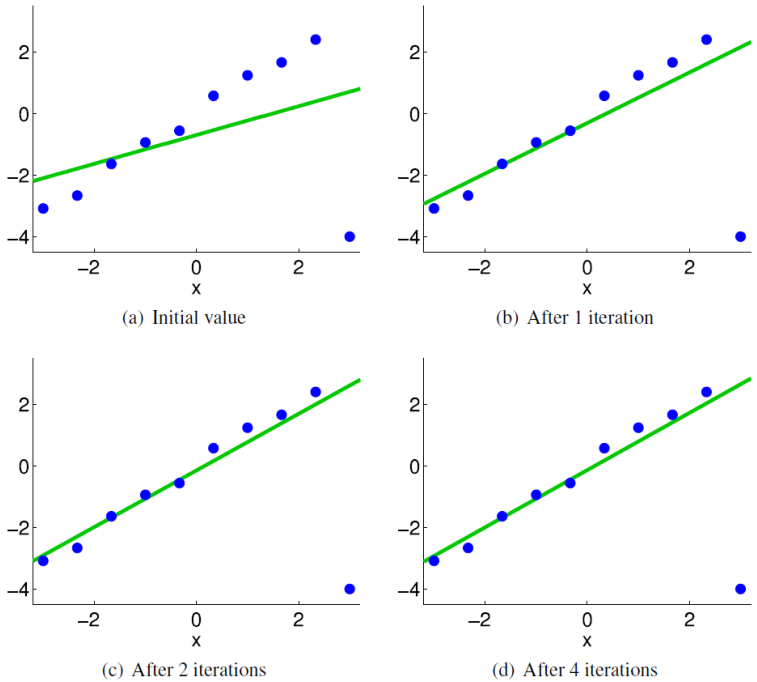
  where $C^{(k)} = \sum_{c_i^{(k)} > \eta} (\frac{1}{2}\eta c_i^{(k)} - \frac{1}{2}\eta^2)$ and $\alpha_i^{(k)} = \begin{cases} 1, & c_i^{(k)} \leqslant \eta \\ \eta / c_i^{(k)}, & c_i^{(k)} > \eta \end{cases}$

- Easy to see that : $J(\mathbf{w}^{(k)}) = \tilde{J}(\mathbf{w}^{(k)}) \geqslant \tilde{J}(\mathbf{w}^{(k+1)}) \geqslant J(\mathbf{w}^{(k+1)})$

Iteratively Reweighted Least Square (IRLS) - Cont'

- This is also called majorization-minimization (MM) algorithm
- Continue the iteration until convergence (guaranteed) to global minimizer (since Huber loss is strictly convex)
- In each step, there is an analytical formula for the weighted least square minimization



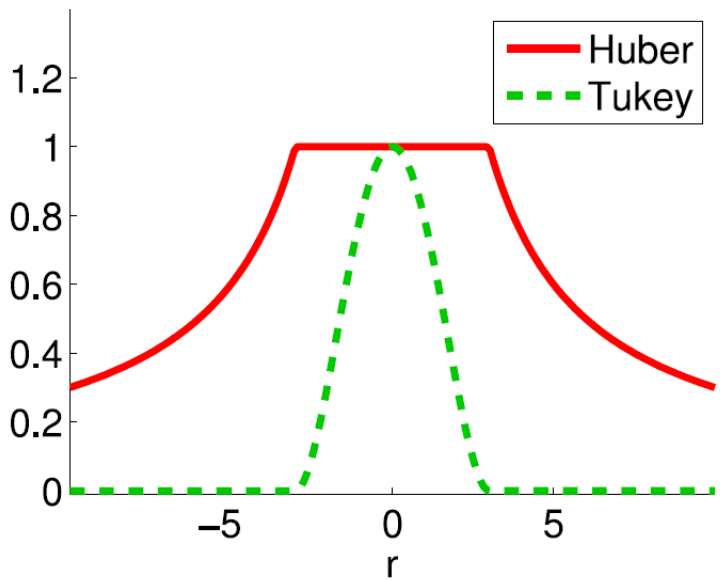


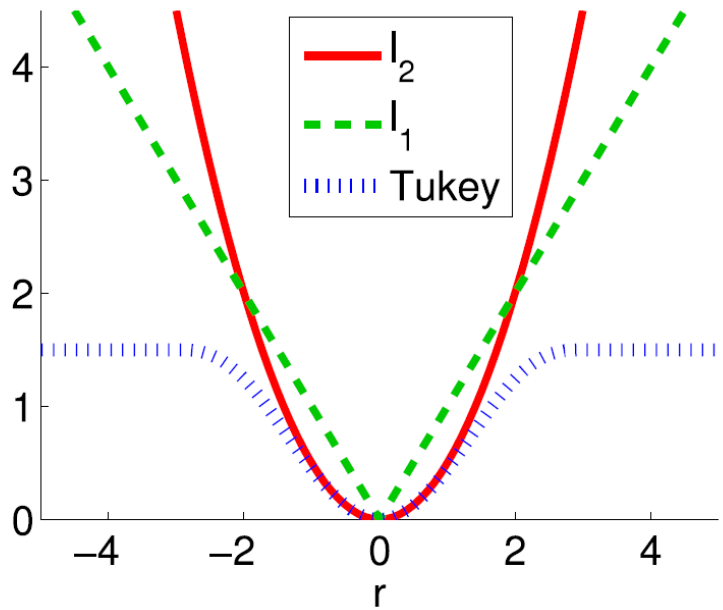
Tukey Loss Minimization

- Further reduce the weights for large residual (left plot)
- Tukey loss (right plot) :

$$\rho_{Tukey}(r) = \begin{cases} \frac{\eta^2}{6}\left(1 - (1 - \frac{r^2}{\eta^2})^3\right), & |r| \leqslant \eta \\ \frac{\eta^2}{6}, & |r| > \eta \end{cases}$$

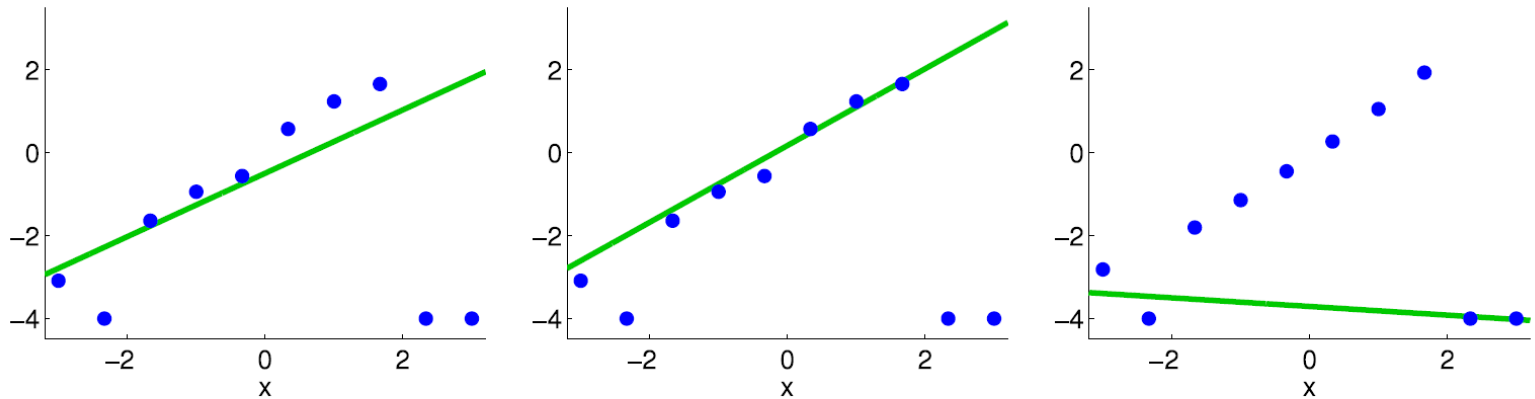
- Can also be solved using (IRLS), but not guaranteed to converge to global minimizer (due to the nonconvexity of Tukey loss)





Other Robust Regressions

- In Tukey regression, slightly changing the noise included in training output samples gives another local optimal solution
- Other candidate losses : pinball loss, deadzone-linear loss, Chebyshev (minimax) approximation, Conditional Value-At-Risk (CVaR) measure, etc.



(a) Huber loss minimization (b) Tukey loss minimization 1 (c) Tukey loss minimization 2

Outlines

Errors and R^2

- Mean absolute error (MAE) : $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- Mean square error (MSE) : $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- Root mean square error (RMSE) : $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- Coefficient of Determination R^2 : $R^2 := 1 - \frac{SS_{res}}{SS_{tot}}$, where $SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y})^2$ is the total sum of squares, and $SS_{res} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ is the residual sum of squares ; $R^2 \in [0, 1]$ (might be negative) ; the larger the R^2, the smaller the ratio of SS_{res} to SS_{tot}, thus the better the model

Adjusted Coefficient of Determination

- Adjusted coefficient of determination : $R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-p-1}$
- n is the number of samples, p is the dimensionality (or the number of attributes)
- The larger the R^2_{adj} value, the better performance the model
- When adding important variables into the model, R^2_{adj} gets larger and SS_{res} is reduced
- When adding unimportant variables into the model, R^2_{adj} may gets smaller and SS_{res} may increase
- In fact, one can show that $1 - R^2_{adj} = \frac{\hat{\sigma}^2}{S^2}$, where $\hat{\sigma}^2 = \frac{1}{n-p-1}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ and $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$ with $(n-p-1)\frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p-1}$ and $(n-1)\frac{S^2}{\sigma^2} \sim \chi^2_{n-1}$ if $\mathbf{w} = \mathbf{0}$.

Outlines

References

- 数据分析导论，博雅大数据学院
- 周志华，机器学习，2016
- T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning : Data mining, Inference, and Prediction, 2nd Edition, 2009