CS 340 Lec. 15: Linear Regression

AD

February 2011

Regression

- Assume you are given some training data $\left\{\mathbf{x}^i, \mathbf{y}^i\right\}_{i=1}^N$ where $x^i \in \mathbb{R}^d$ and $\mathbf{y}^i \in \mathbb{R}^c$.
- Given an input test data $\mathbf{x}$, you want to predict/estimate the output $\mathbf{y}$ associated to $\mathbf{x}$.
- Applications:
  - $\mathbf{x}$: location, y sensor reading.
  - $\mathbf{x}$: stock at time $t-d, t-d+1, ..., t-1$, y: stock at time t.
  - $\mathbf{x}$: temperature at day $t-d, t-d+1, ..., t-1$, y: temperature at day t.

Formalization

- We want to learn a mapping/predictor based on $\left\{\mathbf{x}^i, \mathbf{y}^i\right\}_{i=1}^N$:

$$f : \mathbb{R}^d \to \mathbb{R}^c$$

  which allows us to predict the response $\mathbf{y}$ given a new input $\mathbf{x}$; i.e.

$$\mathbf{y}\left(\mathbf{x}\right) = f\left(\mathbf{x}\right).$$

- Linear regression is the simplest approach to build such a mappnig and is ubiquitous in applied science.

Linear Regression

- For sake of simplicity, consider the simplest case $c = 1$ and $d = 1$.
- Linear regression assumes a model

$$y(x) = w_1 x + w_0$$

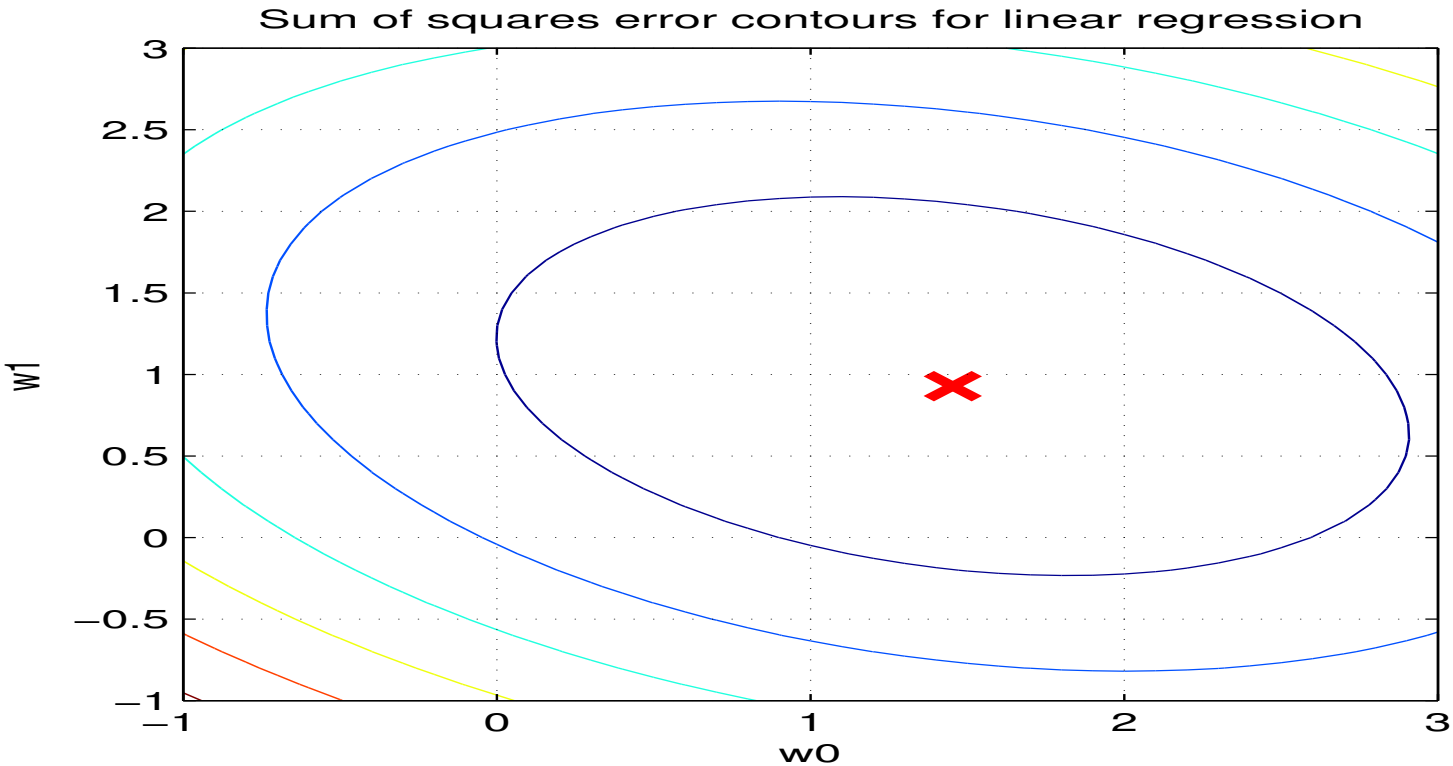
  where $w_1, w_0 \in \mathbb{R}$.
- Given only 2 training data, we can solve for w_1 and w_0 but the result would be dependent of the 2 training data selected and very sensitive to the noise in the training responses y^i.
- A more sensible approach is to minimize the residual errors ε^i over the N training data

$$\begin{aligned} \varepsilon^i &= y^i - y\left(x^i\right) \\ &= y^i - w_1 x^i - w_0 \end{aligned}$$

Least square Regression

- We propose to minimize the sum of the squared residual errors w.r.t (w_0, w_1)

$$E(w_0, w_1) = \sum_{i=1}^{N} \left(y^i - w_1 x^i - w_0\right)^2$$



Least square Regression

- We compute $\frac{\partial E}{\partial w_0}$ and set it equal to zero

$$\begin{aligned} \frac{\partial E}{\partial w_0} &= -2\sum_{i=1}^{N}\left(y^i - w_1 x^i - w_0\right) = 0 \\ \Leftrightarrow \quad w_0 &= \frac{\sum_{i=1}^{N} y^i}{N} - w_1 \frac{\sum_{i=1}^{N} x^i}{N} = \overline{y} - w_1 \overline{x} \end{aligned}$$

- Substituting back $w_0 = \overline{y} - w_1 \overline{x}$ in $E\left(w_0, w_1\right)$, we obtain

$$E\left(\overline{y} - w_1 \overline{x}, w_1\right) = \sum_{i=1}^{N}\left(\left(y^i - \overline{y}\right) - w_1\left(x^i - \overline{x}\right)\right)^2$$

Least square Regression

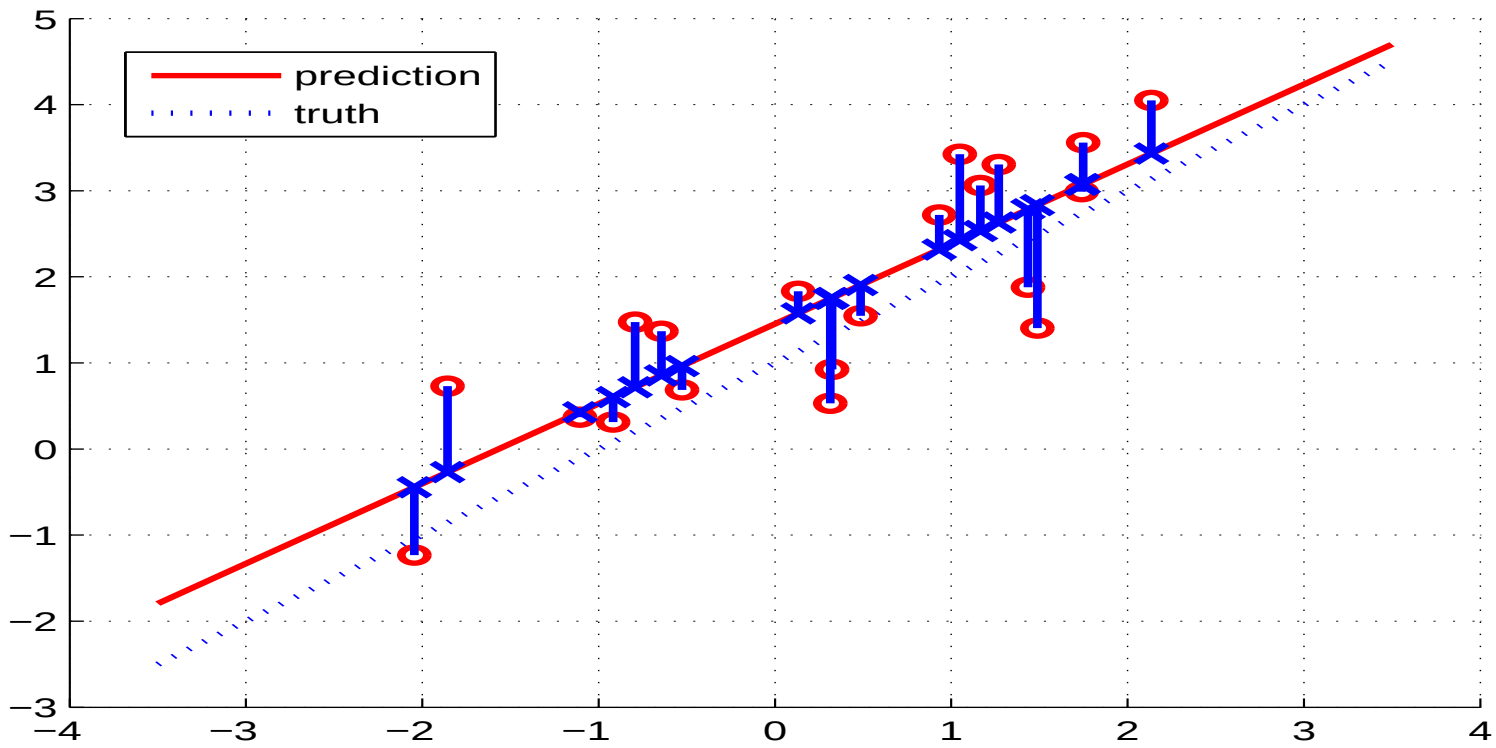
- We compute $\frac{\partial E}{\partial w_1}$ set them equal to zero

$$\begin{aligned}\frac{\partial E}{\partial w_1} &= -2\sum_{i=1}^{N}\left(x^i-\overline{x}\right)\left(\left(y^i-\overline{y}\right)-w_1\left(x^i-\overline{x}\right)\right)=0\\ \Leftrightarrow \quad w_1 &= \frac{\sum_{i=1}^{N}\left(x^i-\overline{x}\right)\left(y^i-\overline{y}\right)}{\sum_{i=1}^{N}\left(x^i-\overline{x}\right)^2}\end{aligned}$$

- Hence

$$w_0=\overline{y}-w_1\overline{x}=\overline{y}-\frac{\sum_{i=1}^{N}\left(x^i-\overline{x}\right)\left(y^i-\overline{y}\right)}{\sum_{i=1}^{N}\left(x^i-\overline{x}\right)^2}\overline{x}.$$

Example



In linear least squares, we minimize the sum of squared distances from each training point (denoted by a red circle) to its approximation (denoted by a blue cross). The red diagonal line represents $y(x) = w_1 x + w_0$, which is the least squares regression line. Note that these residual lines are not perpendicular to the least squares line, in contrast to PCA.

Least square Regression for Multidimensional Inputs

- Consider now that $c = 1$ but $d > 1$ and we consider

$$\begin{aligned} y(\mathbf{x}) &= w_0 + \sum_{k=1}^{d} w_k x_k \\ &= \mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0 = \widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}} \end{aligned}$$

where

$$\begin{aligned} \widetilde{\mathbf{w}}^{\mathsf{T}} &= \begin{pmatrix} w_0 & \mathbf{w}^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} w_0 & w_1 & \cdots & w_d \end{pmatrix} \\ \widetilde{\mathbf{x}}^{\mathsf{T}} &= \begin{pmatrix} 1 & \mathbf{x}^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_1 & \cdots & x_d \end{pmatrix} \end{aligned}$$

- Given N training data, we want to minimize w.r.t $\widetilde{\mathbf{w}}$ the sum of the squared residual errors

$$E(\widetilde{\mathbf{w}}) = \sum_{i=1}^{N} \left(y^i - \widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i \right)^2$$

Least square Regression for Multidimensional Inputs

- We can rewrite

$$E\left(\widetilde{\mathbf{w}}\right)=\left(\mathbf{Y}-\widetilde{\mathbf{X}}\widetilde{\mathbf{w}}\right)^{\mathsf{T}}\left(\mathbf{Y}-\widetilde{\mathbf{X}}\widetilde{\mathbf{w}}\right)$$

where

$$\begin{array}{ll}\mathbf{Y}=\left(y^{1}\ \ldots y^{N}\right)^{\mathsf{T}} & N\times 1 \text{ matrix}\\ \widetilde{\mathbf{X}}=\left(\widetilde{\mathbf{x}}_{1}\ \widetilde{\mathbf{x}}_{2}\ \cdots\ \widetilde{\mathbf{x}}_{N}\right)^{\mathsf{T}} & N\times(d+1)\ \text{ matrix}\end{array}$$

- We can rewrite

$$E\left(\widetilde{\mathbf{w}}\right)=\mathbf{Y}^{\mathsf{T}}\mathbf{Y}-2\mathbf{Y}^{\mathsf{T}}\widetilde{\mathbf{X}}\widetilde{\mathbf{w}}+\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}\widetilde{\mathbf{w}}$$

- By setting $\frac{\partial E}{\partial\widetilde{\mathbf{w}}}=0$, we obtain if $\left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}\right)$ is invertible

$$\widehat{\widetilde{\mathbf{w}}}_{LS}=\left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}\right)^{-1}\widetilde{\mathbf{X}}^{\mathsf{T}}\mathbf{Y}$$

Least square Regression for Multidim Inputs Ouputs

- Consider the case where $c > 1$ and $d > 1$ then

$$\begin{aligned} E\left(\widetilde{\mathbf{w}}\right) &= \sum_{i=1}^{N} \text{trace}\left(\mathbf{y}^i - \widetilde{\mathbf{W}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i\right)^{\mathsf{T}}\left(\mathbf{y}^i - \widetilde{\mathbf{W}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i\right) \\ &= \left\|\mathbf{Y} - \widetilde{\mathbf{X}}\widetilde{\mathbf{W}}\right\|_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{c}\left(y_j^i - \left(\widetilde{\mathbf{W}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i\right)_j\right)^2 \end{aligned}$$

  where $\widetilde{\mathbf{W}} \in \mathbb{R}^{(d+1)\times c}$.

- It can also be shown that

$$\widehat{\widetilde{\mathbf{w}}}_{LS} = \left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}\right)^{-1}\widetilde{\mathbf{X}}^{\mathsf{T}}\mathbf{Y}$$

Geometric Interpretation

- For sake of simplicity, let's come back to the case where $c = 1$. We have for an new input $\mathbf{x}$

$$y(\mathbf{x}) = \widehat{\widetilde{\mathbf{w}}}_{LS}^{\mathsf{T}} \widetilde{\mathbf{x}} = \widetilde{\mathbf{x}}^{\mathsf{T}} \widehat{\widetilde{\mathbf{w}}}_{LS} = \widetilde{\mathbf{x}}^{\mathsf{T}} \left(\widetilde{\mathbf{X}}^{\mathsf{T}} \widetilde{\mathbf{X}} \right)^{-1} \widetilde{\mathbf{X}}^{\mathsf{T}} \mathbf{Y}$$

- On the training set $\left\{ \mathbf{x}^i, \mathbf{y}^i \right\}_{i=1}^N$, we have

$$\mathbf{Y}(\mathbf{X}) = \widetilde{\mathbf{X}} \widehat{\widetilde{\mathbf{w}}}_{LS} = \widetilde{\mathbf{X}} \left(\widetilde{\mathbf{X}}^{\mathsf{T}} \widetilde{\mathbf{X}} \right)^{-1} \widetilde{\mathbf{X}}^{\mathsf{T}} \mathbf{Y}$$

so

$$\begin{aligned} \widetilde{\mathbf{X}}^{\mathsf{T}} \left(\mathbf{Y}(\mathbf{X}) - \mathbf{Y} \right) &= \widetilde{\mathbf{X}}^{\mathsf{T}} \widetilde{\mathbf{X}} \left(\widetilde{\mathbf{X}}^{\mathsf{T}} \widetilde{\mathbf{X}} \right)^{-1} \widetilde{\mathbf{X}}^{\mathsf{T}} \mathbf{Y} - \widetilde{\mathbf{X}}^{\mathsf{T}} \mathbf{Y} \\ &= 0 \end{aligned}$$

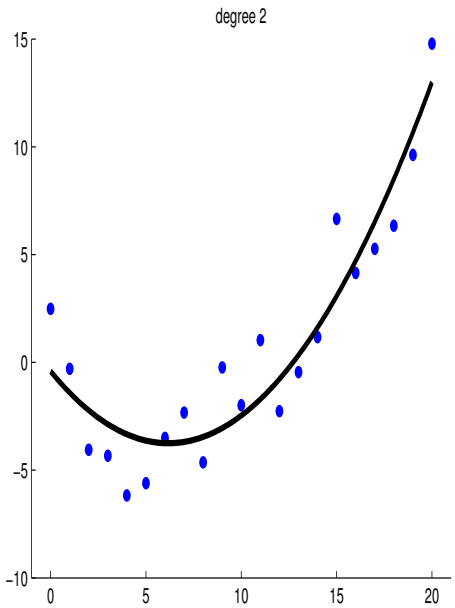
- $\mathbf{Y}(\mathbf{X})$ is simply the orthogonal projection of $\mathbf{Y}$ onto the columns of $\widetilde{\mathbf{X}}$.

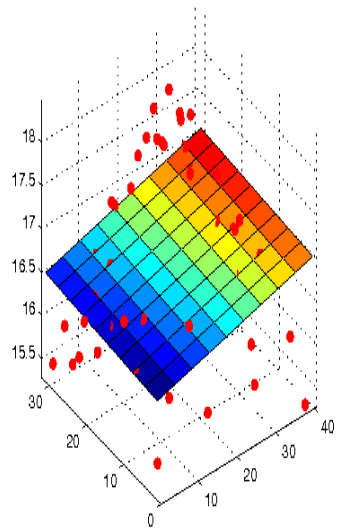
Nonlinear Regression using Basis Functions

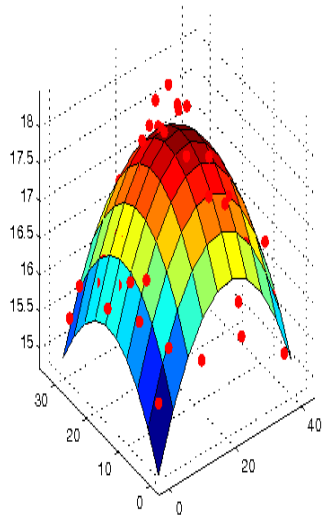
- Linear regression can handle nonlinear regression problem; e.g.

$$y(\mathbf{x}) = \widetilde{\mathbf{w}}^{\mathsf{T}} \Phi(\widetilde{\mathbf{x}})$$

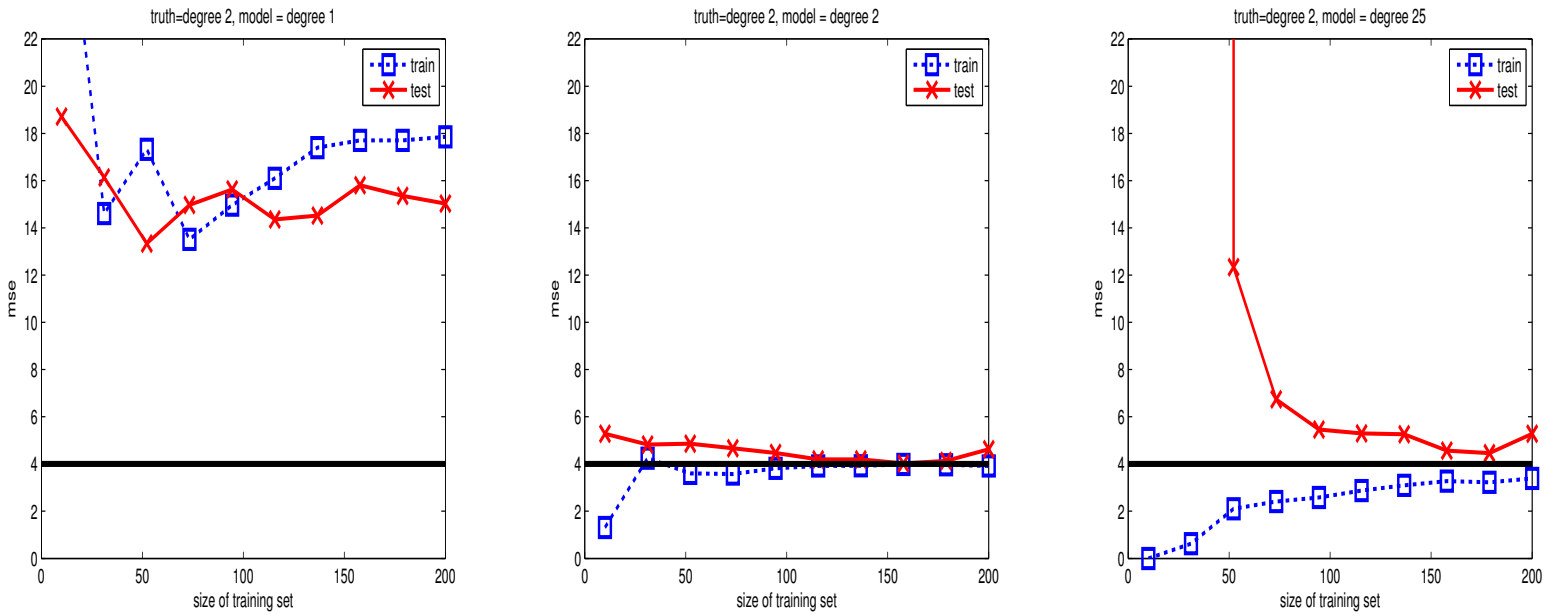
  where $\Phi : \mathbb{R}^{d+1} \longrightarrow \mathbb{R}^{m}$ and $\widetilde{\mathbf{w}}$ is a $m-$dimensional vector.
- Example: $\Phi(\widetilde{\mathbf{x}}) = \left(1, x, x^2\right)^{\mathsf{T}}$ $(d = 1, m = 3)$; $\Phi(\widetilde{\mathbf{x}}) = \widetilde{\mathbf{x}} = (1, x_1, x_2)$ $(d = 2, m = 3)$; $\Phi(\widetilde{\mathbf{x}}) = \widetilde{\mathbf{x}} = \left(1, x_1, x_2, x_1^2, x_2^2\right)$ $(d = 2, m = 5)$.







MSE on Training and Test Sets



Data generated from a degree 2 poly. with Gaussian noise of var. 4. We fit polynomial models. For N small, test error of the degree 25 poly. is higher than that of the degree 2 poly., due to overfitting, but this difference vanishes once as N increases. Note also that the degree 1 poly. is too simple and has high test error even given large N.

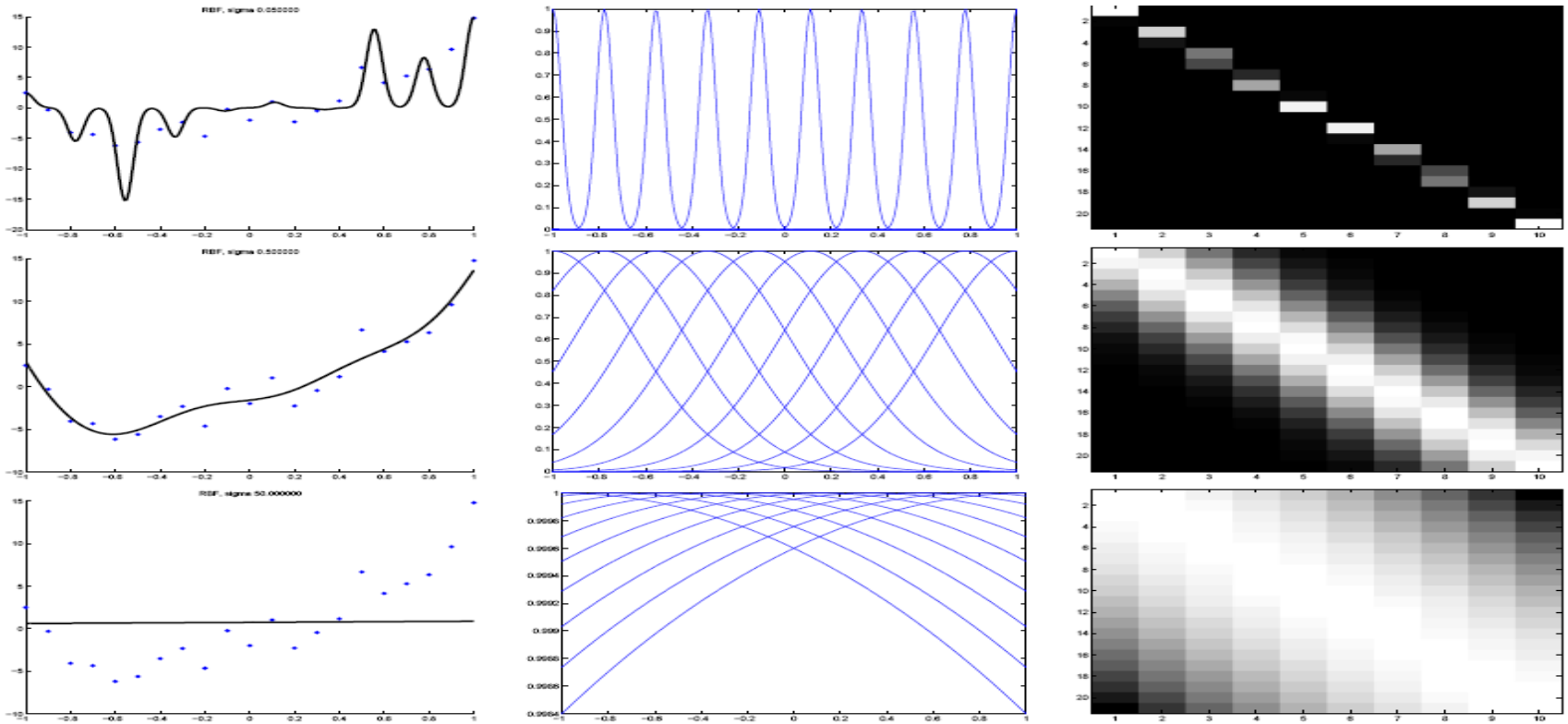
Kernel Regression

- Another way to perform nonlinear regression is to use kernels to define the basis functions $\Phi(\mathbf{x}) = (K(\mathbf{x}, \boldsymbol{\mu}_1), ..., K(\mathbf{x}, \boldsymbol{\mu}_m))$.
- For example, we could use a Radial Basis Function

$$K(\mathbf{x}, \boldsymbol{\mu}) = \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}(\mathbf{x}-\boldsymbol{\mu})}{2\sigma^2}\right).$$

- Alternatively we can use any function: wavelets, curvelets, splines etc.
- Selecting $(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_m, \sigma^2)$ can be difficult.
- How to select $\boldsymbol{\mu}$: 1) place the centers unformly spaced in the region containing the data, 2) place one center at each data point, 3) cluster the data and use one center for each cluster, 4) use CV, MLE or Bayesian approach.
- How to select σ^2 : 1) use average squared distances to neighboring centers (scaled by a constant), 2) use CV, MLE or Bayesian approach.

Kernel Regression



(Left) RBS basis in 1d. (Middle) Basis functions evaluated on a grid. (Right) Design matrix.

A Probabilistic Interpretation of Least-Square Regression

- Consider the following model

$$p\left(y|\,\mathbf{x},\widetilde{\mathbf{w}},\sigma^2\right)=\mathcal{N}\left(y;y\left(\mathbf{x}\right),\sigma^2\right)$$

where $y\left(\mathbf{x}\right)=\widetilde{\mathbf{w}}^{\mathsf{T}}\Phi\left(\widetilde{\mathbf{x}}\right)$.

- Given the training set $\left\{\mathbf{x}^i,\mathbf{y}^i\right\}_{i=1}^N$, the MLE estimates of $\left(\widetilde{\mathbf{w}},\sigma^2\right)$ is given by

$$\left(\widehat{\widetilde{\mathbf{w}}}_{MLE},\widehat{\sigma^2}_{MLE}\right)=\underset{\left(\widetilde{\mathbf{w}},\sigma^2\right)}{\arg\max}\sum_{i=1}^{N}\log p\left(y^i\,\middle|\,\mathbf{x}^i,\widetilde{\mathbf{w}},\sigma^2\right)$$

where

$$\sum_{i=1}^{N}\log p\left(y^i\,\middle|\,\mathbf{x}^i,\widetilde{\mathbf{w}},\sigma^2\right)=-\frac{N}{2}\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y^i-\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i\right)^2$$

- Hence $\widehat{\widetilde{\mathbf{w}}}_{MLE}=\left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}\right)^{-1}\widetilde{\mathbf{X}}^{\mathsf{T}}\mathbf{Y}$ and $\widehat{\sigma^2}_{MLE}=\sum_{i=1}^{N}\left(y^i-\widehat{\widetilde{\mathbf{w}}}_{MLE}\widetilde{\mathbf{x}}^i\right)^2/N$.

Robust Regression

- The problem with least square regression is that it is very sensitive to outliers as it minimizes

$$E_{L2}\left(\widetilde{\mathbf{w}}\right)=\sum_{i=1}^{N}\left(y^{i}-y\left(\mathbf{x}^{i}\right)\right)^{2}=\sum_{i=1}^{N}\left(y^{i}-\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{x}}^{i}\right)^{2}$$

  and hence the square of the residual errors.
- To design a procedure less sensitive to these outliers, we could pick

$$E_{L1}\left(\widetilde{\mathbf{w}}\right)=\sum_{i=1}^{N}\left|y^{i}-y\left(\mathbf{x}^{i}\right)\right|=\sum_{i=1}^{N}\left|y^{i}-\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{x}}^{i}\right|.$$

- As $|u|$ goes to infinity slowler than u^2 as $|u|\rightarrow\infty$ then this procedure is less sensitive to outliers. We could also use something like $E_{Huber}\left(\widetilde{\mathbf{w}}\right)=\sum_{i=1}^{N}c\left(y^{i}-y\left(\mathbf{x}^{i}\right)\right)$ where

$$c\left(u\right)=\begin{cases}|u| & \text{if } |u|\leq|u_{0}|\\ |u_{0}| & \text{if } |u|\geq|u_{0}|\end{cases}$$

A Probabilistic Interpretation of Robust Regression

- Consider the following model

$$p\left(y\middle|\,\mathbf{x},\widetilde{\mathbf{w}},b\right)=\text{Lap}\left(y;y\left(\mathbf{x}\right),b\right)$$

where $y\left(\mathbf{x}\right)=\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}$ and

$$\text{Lap}\left(x;\mu,b\right)=\frac{1}{2b}\exp\left(-\frac{\left|x-\mu\right|}{b}\right)$$

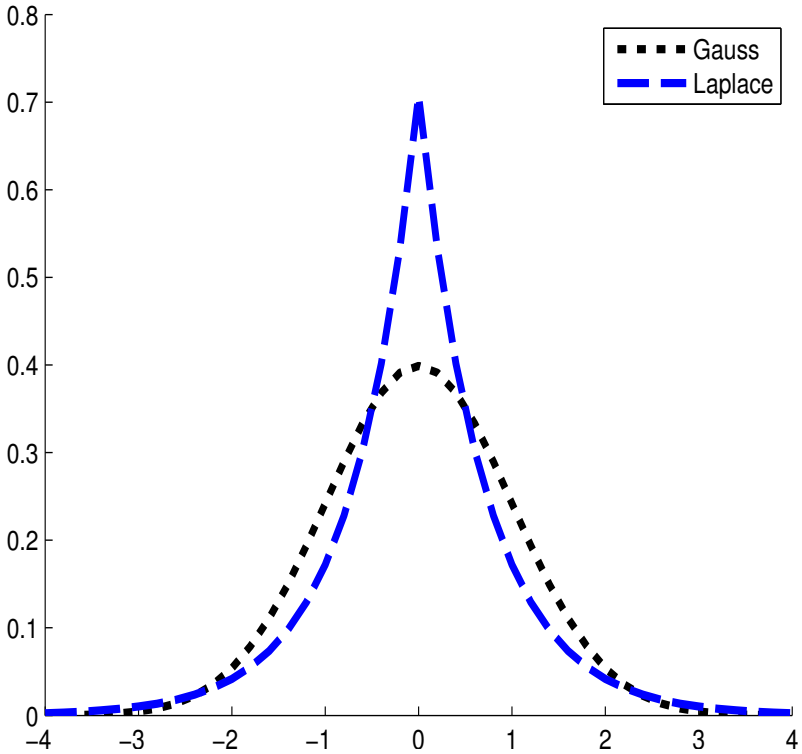
is the Laplace density of location μ and scale b.

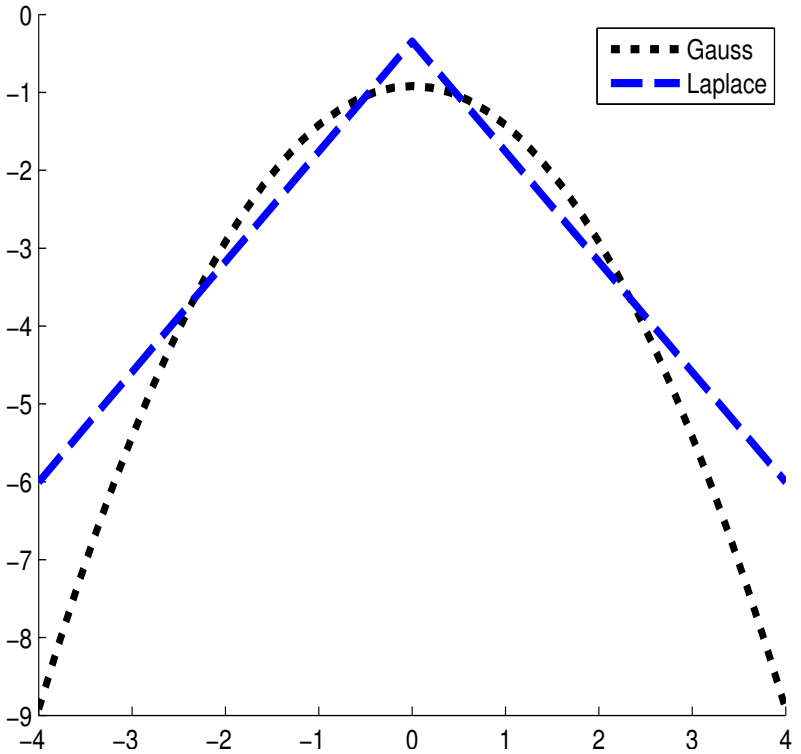
- Given the training set $\left\{\mathbf{x}^i,\mathbf{y}^i\right\}_{i=1}^N$, the MLE estimates of $\left(\widetilde{\mathbf{w}},\sigma^2\right)$ is given by

$$\begin{aligned}\left(\widehat{\widetilde{\mathbf{w}}}_{MLE,Laplace},\widehat{b}_{MLE}\right) &= \underset{(\widetilde{\mathbf{w}},b)}{\arg\max}\sum_{i=1}^{N}\log p\left(y^i\middle|\,\mathbf{x}^i,\widetilde{\mathbf{w}},b\right)\\ &= \arg\max\ -N\log\left(2b\right)-\frac{1}{b}\sum_{i=1}^{N}\left|y^i-\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i\right|.\end{aligned}$$

- That is $\widehat{\widetilde{\mathbf{w}}}_{MLE,Laplace}$ minimizes $E_{L1}\left(\widetilde{\mathbf{w}}\right)$.

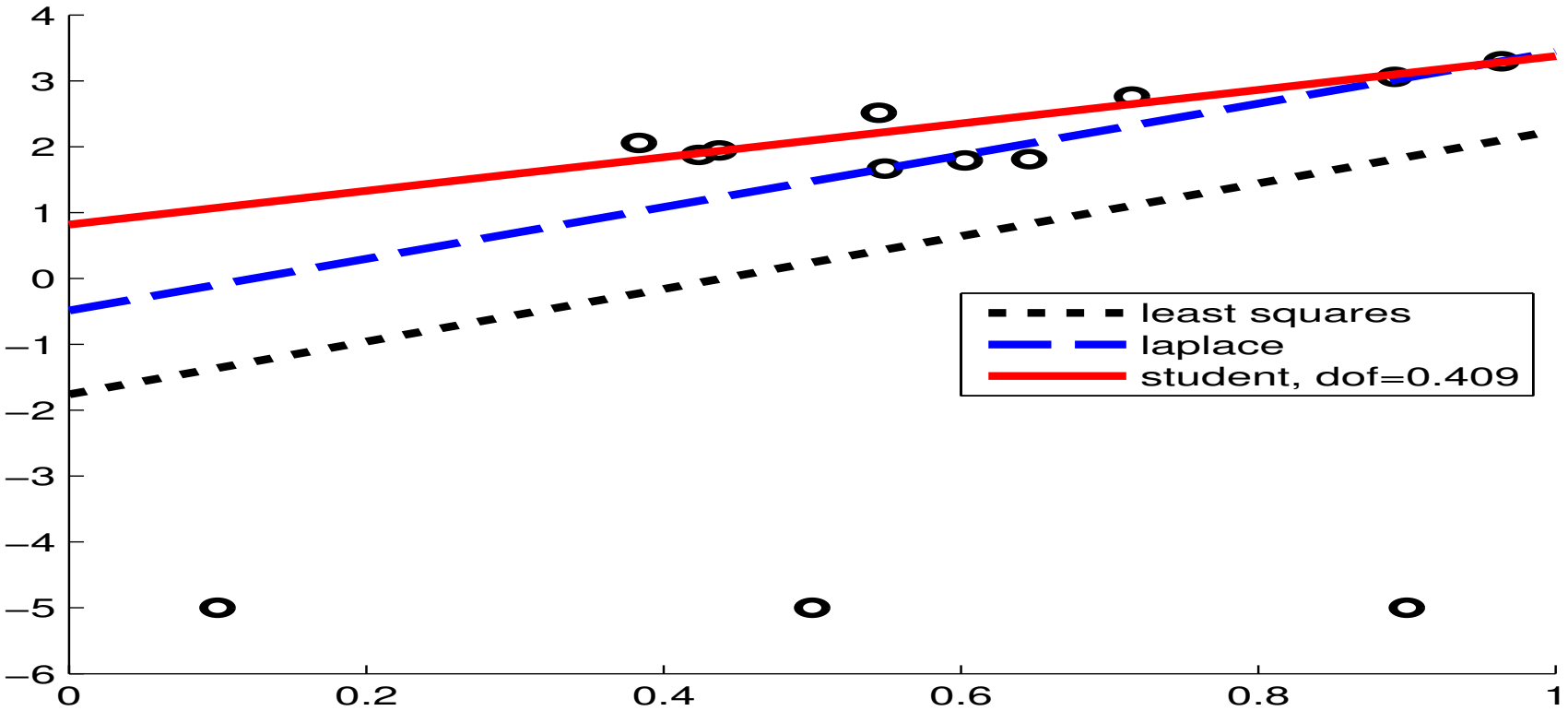
Gauss versus Laplace





(left) Plots of $\mathcal{N}(y; 0, 1)$ and $\text{Lap}\left(y; 0, 1/\sqrt{2}\right)$ densities (both have zero-mean unit variance). (right) Negative logs of these pdfs.

Gauss versus Laplace Regression



Linear regression with outliers. Data points are black circles. Black dottet line: LS, Blue dashed line: Laplace.

Limitations of Least-Square Regression

- Remember that our estimate is given by

$$\widehat{\widetilde{\mathbf{w}}}_{LS} = \left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}\right)^{-1}\widetilde{\mathbf{X}}^{\mathsf{T}}\mathbf{Y}$$

- This assumes that the $(d+1)\times(d+1)$ matrix $\left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}\right)$ is invertible.
- We have $\text{rank}\left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}\right) = \text{rank}\left(\widetilde{\mathbf{X}}\right) \leq \min\left(N, d+1\right)$. Hence in common scenarios where $N < d+1$, we can never invert $\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}$!
- We have also problems when columns/rows of $\widetilde{\mathbf{X}}$ are almost linearly dependent (collinearity).

The Need for Regularization

- When we have a small number of data, we want to be able to regularize the solution and limit overfitting.
- When fitting a polynomial regression model, "wiggly" functions will have large weights $\widetilde{\mathbf{w}}$.
- For example for the 14 polynomial model fitted previously, we have 11 coefficients w_k such that $|\widehat{w}_{k,LS}| > 100$!

Ridge Regression

- To bypass the fact that $\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}$ might not be invertible, we consider instead

$$\widehat{\widetilde{\mathbf{w}}}_R = \left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}} + \lambda\ I_{d+1}\right)^{-1}\widetilde{\mathbf{X}}^{\mathsf{T}}\mathbf{Y}$$

  where $\lambda > 0$ and I_{d+1} is the $(d+1)$ identity matrix.
- This estimate minimizes

$$E\left(\widetilde{\mathbf{w}}\right) = \sum_{i=1}^{N}\left(y^i - \widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i\right)^2 + \lambda\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{w}}.$$

- This is equivalent to minimize

$$\sum_{i=1}^{N}\left(y^i - \widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i\right)^2 \quad \text{s.t. } \widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{w}} \leq t\left(\lambda\right)$$

- This shrinks the value of $\widehat{\widetilde{\mathbf{w}}}$ towards zeros.

A Probabilistic Interpretation of Ridge Regression

- Consider the following model

$$p\left(y|\,\mathbf{x},\widetilde{\mathbf{w}},\sigma^2\right)=\mathcal{N}\left(y;y\left(\mathbf{x}\right),\sigma^2\right)$$

where $y\left(\mathbf{x}\right)=\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}$ and

$$p\left(\widetilde{\mathbf{w}}|\,\sigma^2\right)=\prod_{k=0}^{d}\mathcal{N}\left(w_k;0,\sigma^2/\lambda\right)$$

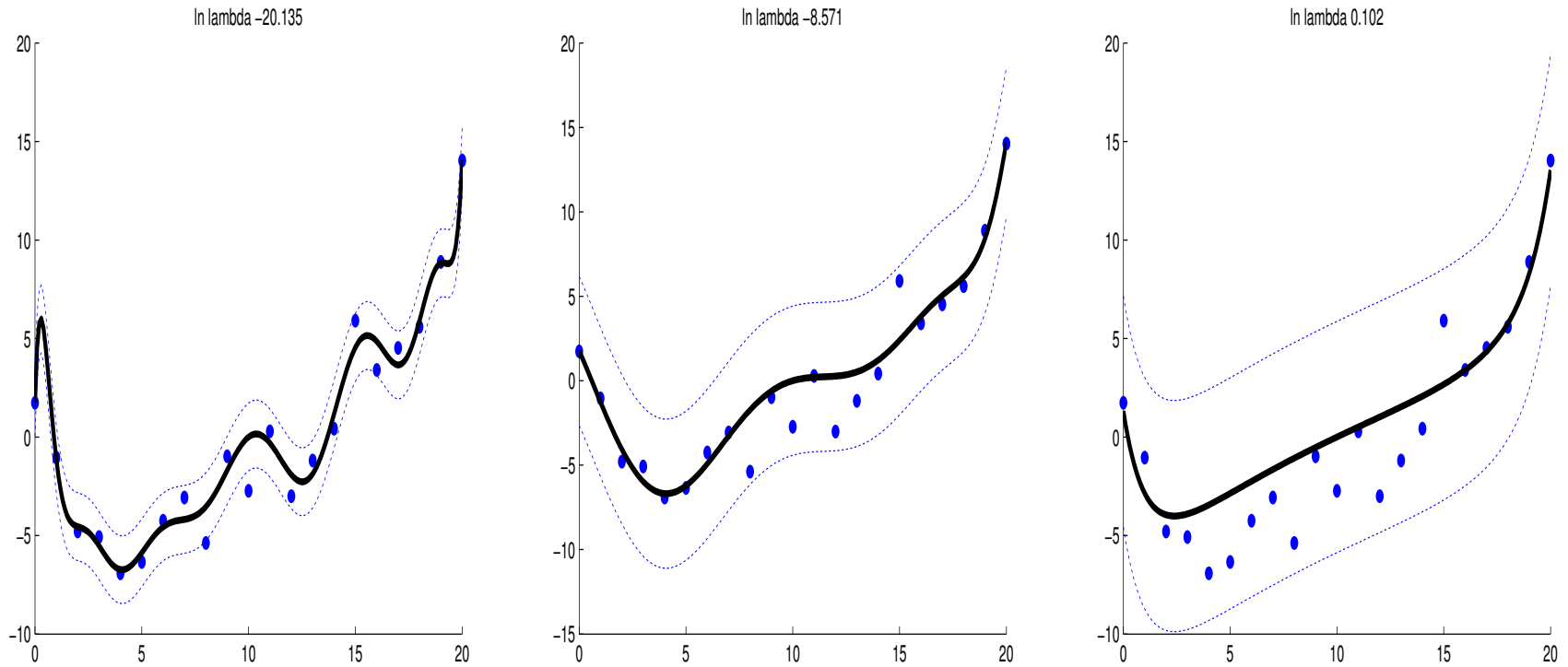
- In practice, we usually set a flat prior on w_0; i.e $\mathcal{N}\left(w_0;0,\sigma^2/\lambda_0\right)$ with $\lambda_0<<1$.
- Given $\left\{\mathbf{x}^i,\mathbf{y}^i\right\}_{i=1}^N$, the MAP estimate of $\widetilde{\mathbf{w}}$ is given by

$$\widehat{\widetilde{\mathbf{w}}}_{MAP}=\arg\max\ \log p\left(\widetilde{\mathbf{w}}|\,\mathbf{x}^{1:N},y^{1:N},\sigma^2\right)$$

where
$\log p\left(\widetilde{\mathbf{w}}|\,\mathbf{x}^{1:N},y^{1:N},\sigma^2\right)=\sum_{i=1}^{N}\log p\left(y^i\left|\,\mathbf{x}^i,\widetilde{\mathbf{w}},\sigma^2\right.\right)+\log p\left(\widetilde{\mathbf{w}}|\,\sigma^2\right)=-\frac{N}{2}\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y^i-\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}^i\right)^2-\frac{\lambda}{2\sigma^2}\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{w}}$

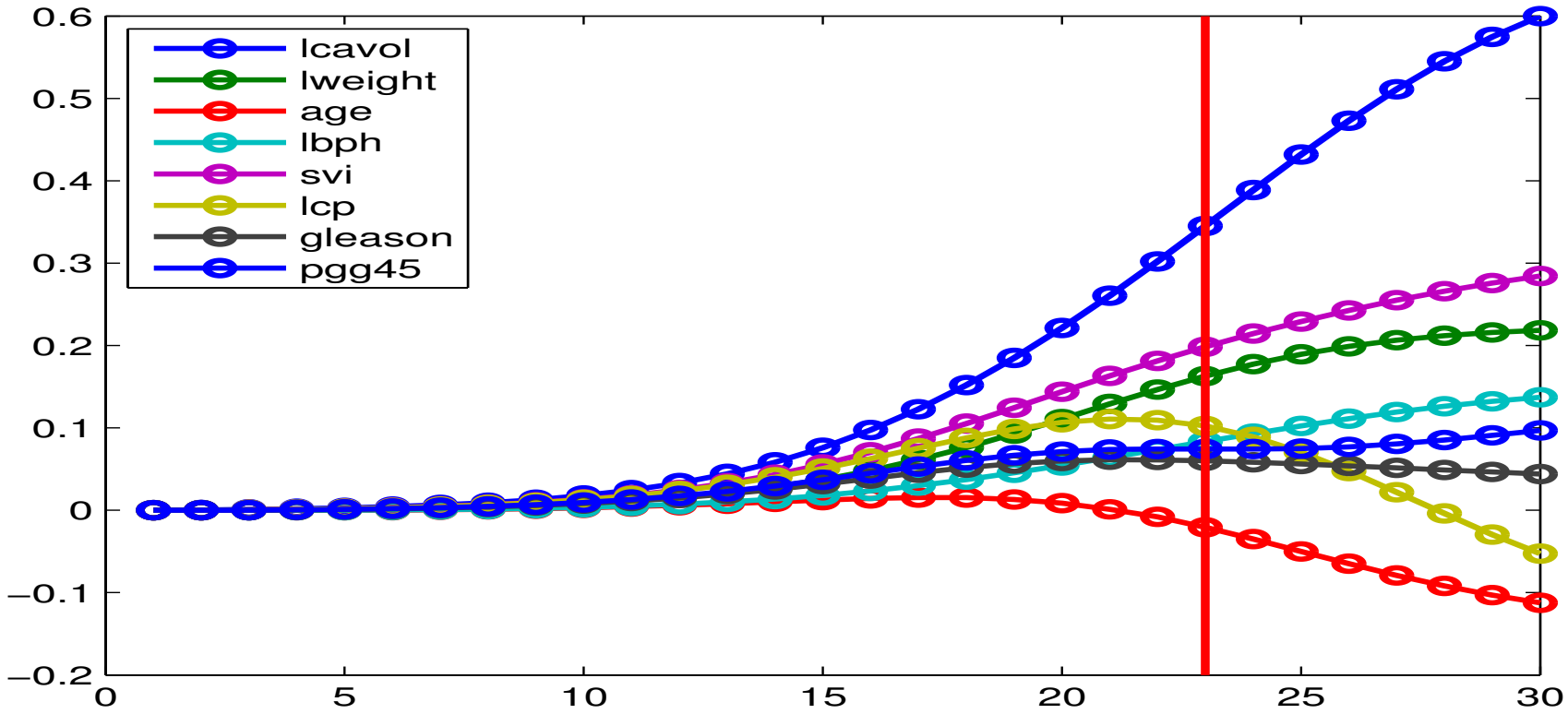
- Hence $\widehat{\widetilde{\mathbf{w}}}_{MAP}=\widehat{\widetilde{\mathbf{w}}}_R=\left(\widetilde{\mathbf{X}}^{\mathsf{T}}\widetilde{\mathbf{X}}+\lambda\ I_{d+1}\right)^{-1}\widetilde{\mathbf{X}}^{\mathsf{T}}\mathbf{Y}$.

Example of Ridge Regression



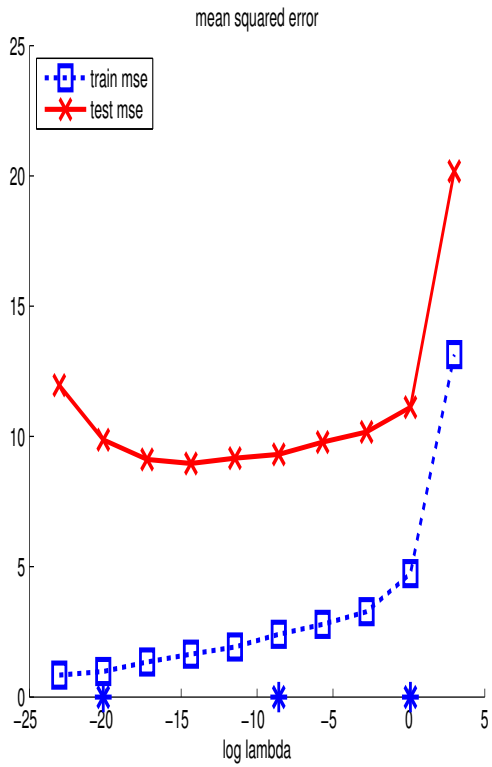
Degree 14 polynomial fit to $N = 21$ data points with increasing λ. The errors bars, represents the standard deviation $\widehat{\sigma}$.

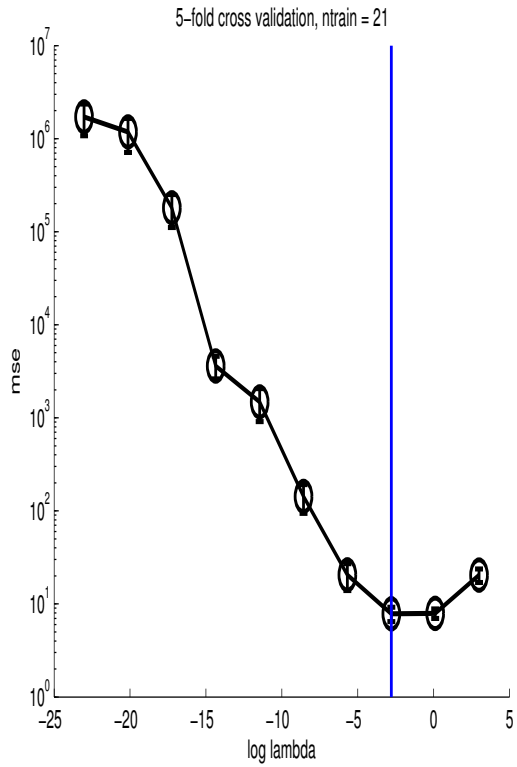
Regularization Path for Ridge Regression

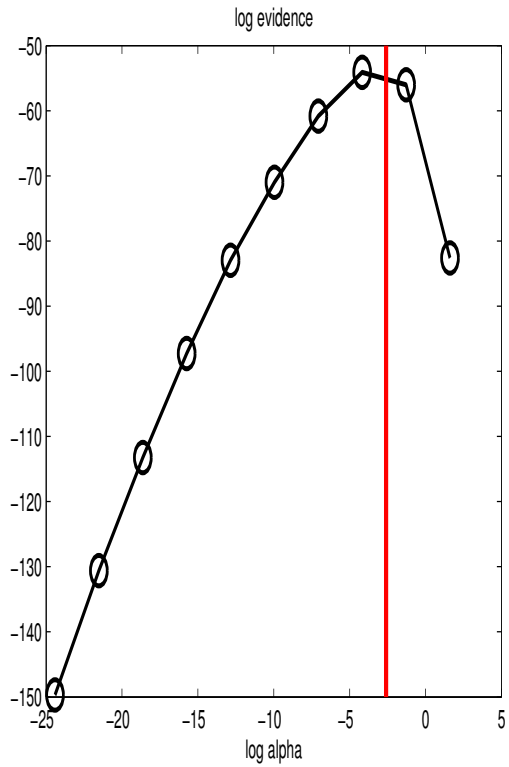


Profile of Ridge coefficients for an example on real data where $d = 8$ vs bound on $\mathbf{w}^{\mathsf{T}}\mathbf{w}$, i.e. small $t(\lambda)$ means large λ.

Example of Ridge Regression







(left) Training and test error for a degree 14 poly. with increasing λ. (center) Estimate of test MSE produced by 5-fold CV. (right) Log-marginal likelihood vs $\log(\alpha)$ where $\alpha = \lambda/\sigma^2$.

L1 Regression - Lasso

- We minimize in this case

$$E\left(\widetilde{\mathbf{w}}\right)=\sum_{i=1}^{N}\left(y^{i}-\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}^{i}\right)^{2}+\lambda\left\|\widetilde{\mathbf{w}}\right\|$$

where $\left\|\widetilde{\mathbf{w}}\right\|=\sum_{k=0}^{d}\left|w_{k}\right|$. This is equivalent to minimize

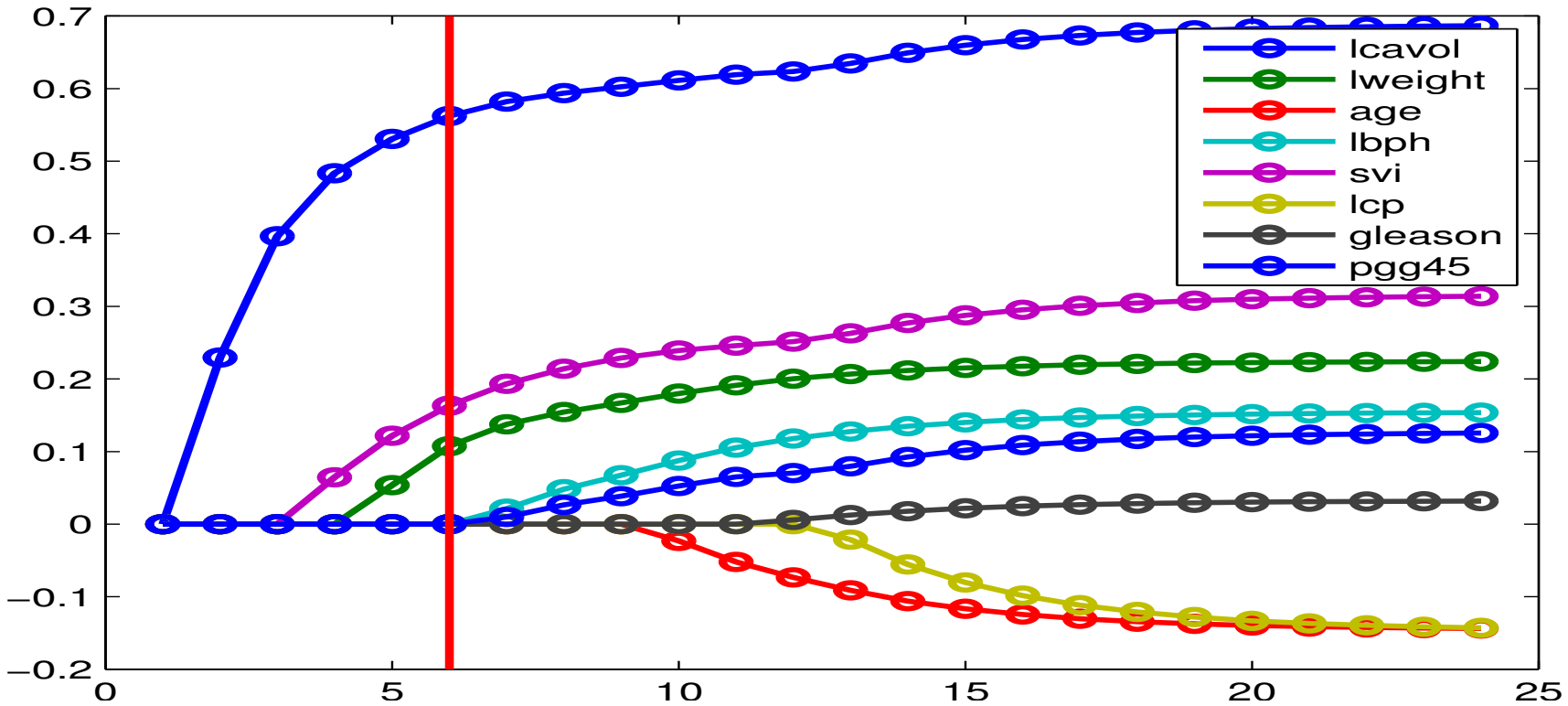
$$\sum_{i=1}^{N}\left(y^{i}-\widetilde{\mathbf{w}}^{\mathsf{T}}\widetilde{\mathbf{x}}^{i}\right)^{2}\ \text{s.t.}\ \left\|\widetilde{\mathbf{w}}\right\|\leq t\left(\lambda\right)$$

- Given $\left\{\mathbf{x}^{i},\mathbf{y}^{i}\right\}_{i=1}^{N}$, this is equivalent to taking the MAP estimate of $\widetilde{\mathbf{w}}$ associated to

$$p\left(y|\,\mathbf{x},\widetilde{\mathbf{w}},\sigma^{2}\right)=\mathcal{N}\left(y;y\left(\mathbf{x}\right),\sigma^{2}\right),$$

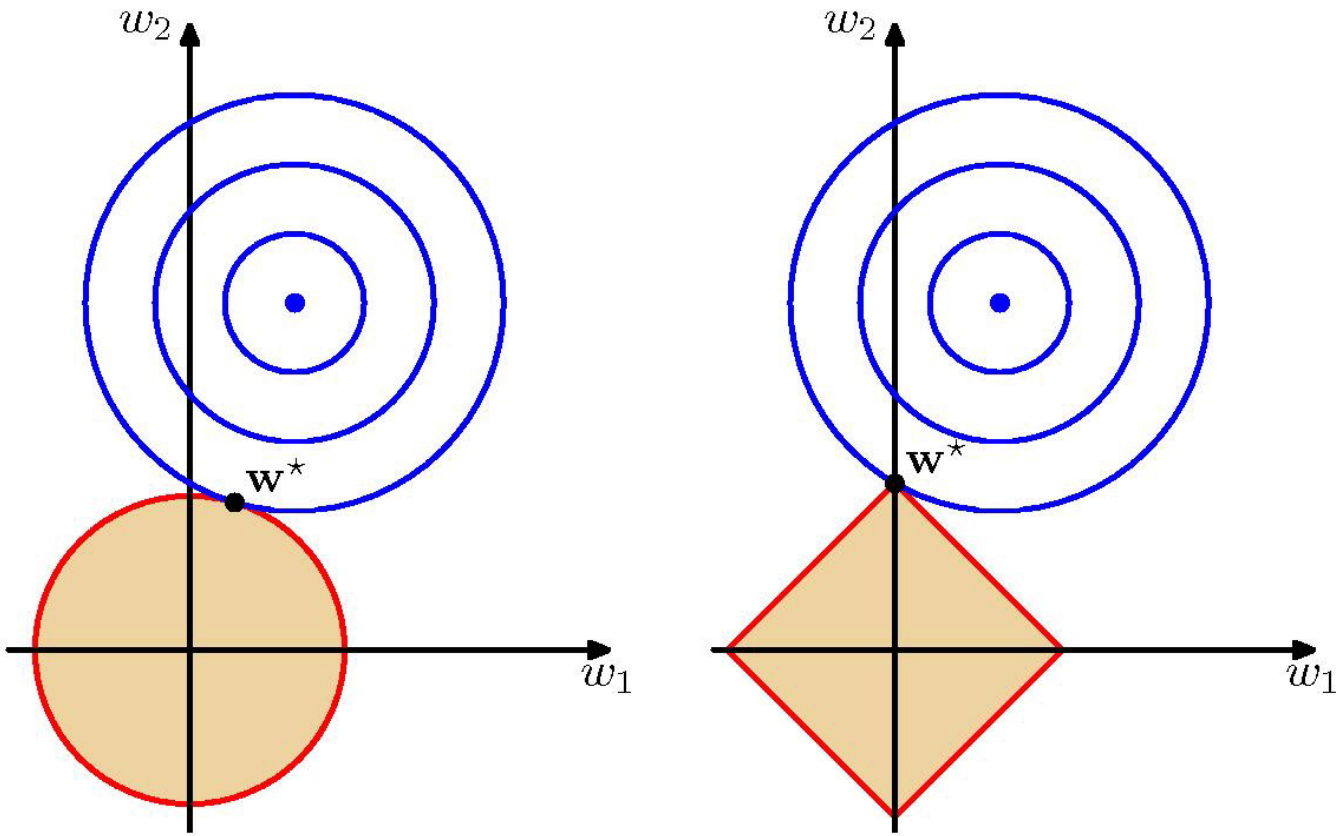
$$p\left(\widetilde{\mathbf{w}}|\,\sigma^{2}\right)=\prod_{k=0}^{d}\mathrm{Lap}\left(w_{k};0,\sigma^{2}/\lambda\right).$$

Regularization Path for Lasso



Profile of Lasso coefficients for an example on real data where $d = 8$ vs bound on $\sum_{k=1}^{d} |w_k|$, i.e. small $t(\lambda)$ means large λ.

Ridge versus Lasso



Contours of the unregularized function (blue) along with the constraint: ridge (left) and lasso (right). The lasso give a sparse solution as $w_1^* = 0$. The corners of the simplex is more likely to intersect the ellipse than one of the sides as they "stick out" more.