



So Far...

- ▶ Our goal (supervised learning):
 - To learn a set of discriminant functions

- ▶ Bayesian framework
 - We could design an optimal classifier if we knew:
 - $P(\omega_i)$: priors and $P(x | \omega_i)$: class-conditional densities
 - Using training data to estimate $P(\omega_i)$ and $P(x | \omega_i)$
 - $P(\omega_i | x)$ is computed and be used as the discriminant functions

- ▶ Other possible ways?
 - Directly learning discriminant functions from the training data

Linear Methods for Regression

Deng Cai (蔡登)

College of Computer Science
Zhejiang University

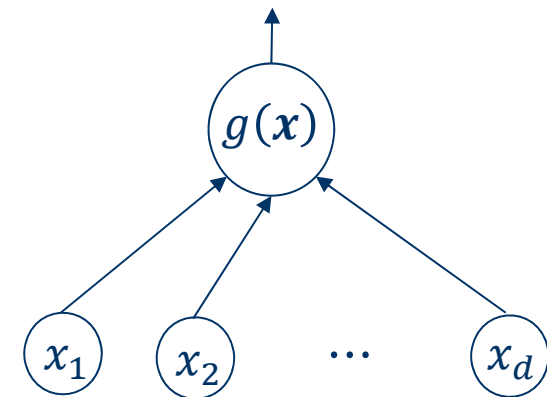
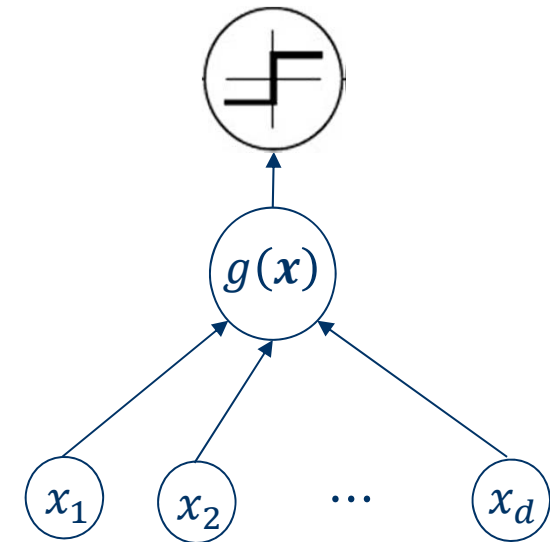
dengcai@gmail.com





Classification VS Regression

- ▶ Both are supervised learning methods
 - Goal: learn a mapping from inputs \mathbf{x} to outputs y
- ▶ Classification (Categorization, Decision making...)
 - y is a categorical variable
- ▶ Regression
 - y is real-valued





Linear model

- ▶ Sample: $\mathbf{x} \in R^d, \mathbf{x} = [x_1, x_2, \dots, x_d]^T$
- ▶ Finds a linear function $\mathbf{w} = [w_1, w_2, \dots, w_d]^T \in R^d, b$

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

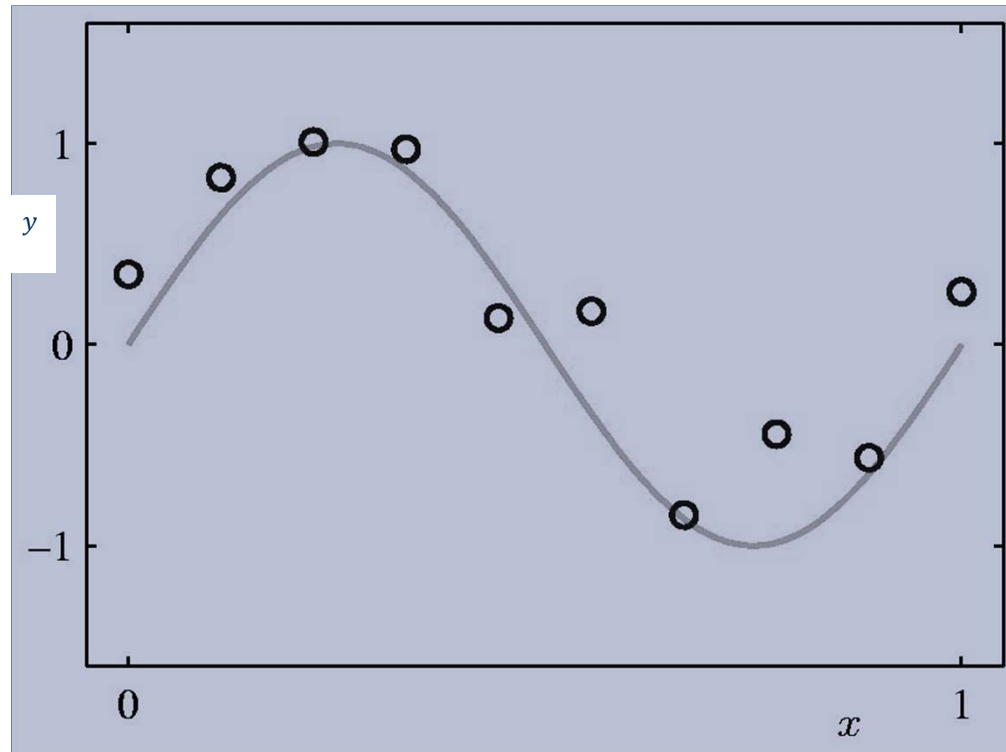
$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

$$\mathbf{x} = [x_1, x_2, \dots, x_d, 1]^T \in R^{d+1}$$

$$\mathbf{w} = [w_1, w_2, \dots, w_d, b]^T \in R^{d+1}$$



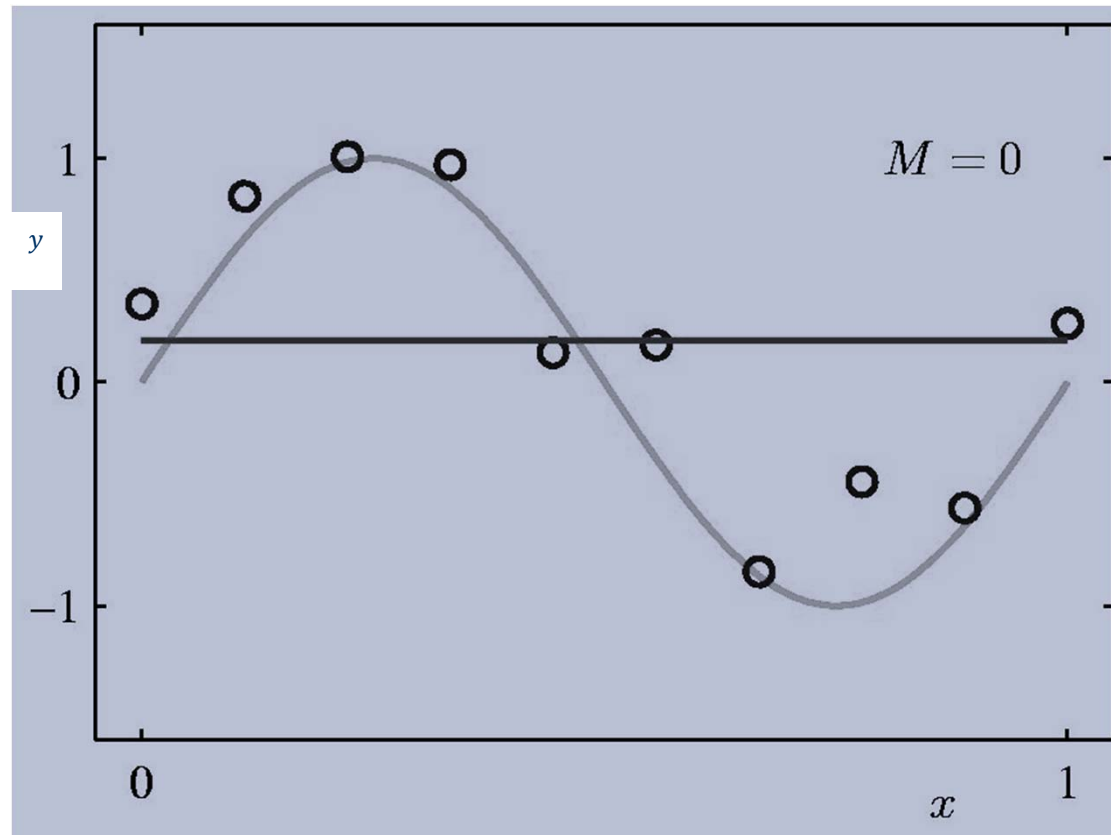
Polynomial Curve Fitting



$$f(x, \mathbf{a}) = a_0 + a_1x + a_2x^2 + \cdots + a_Mx^M = \sum_{j=0}^M a_jx^j$$

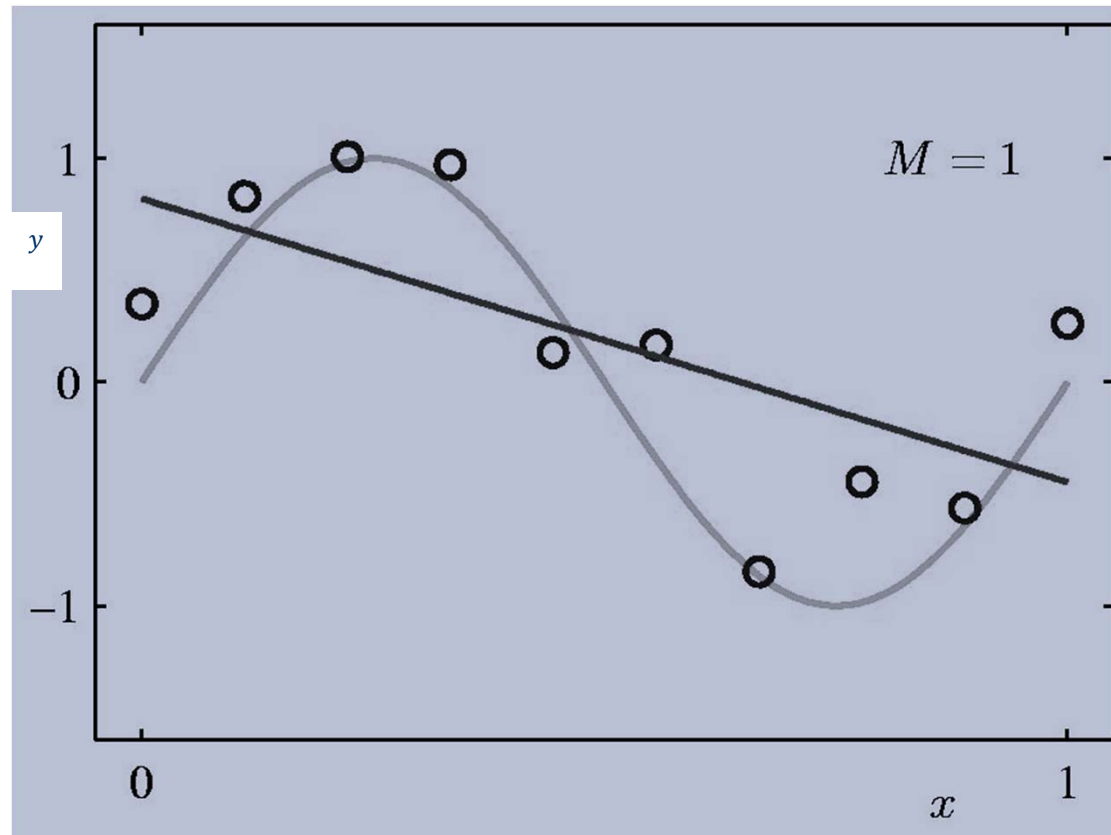


0th Order Polynomial



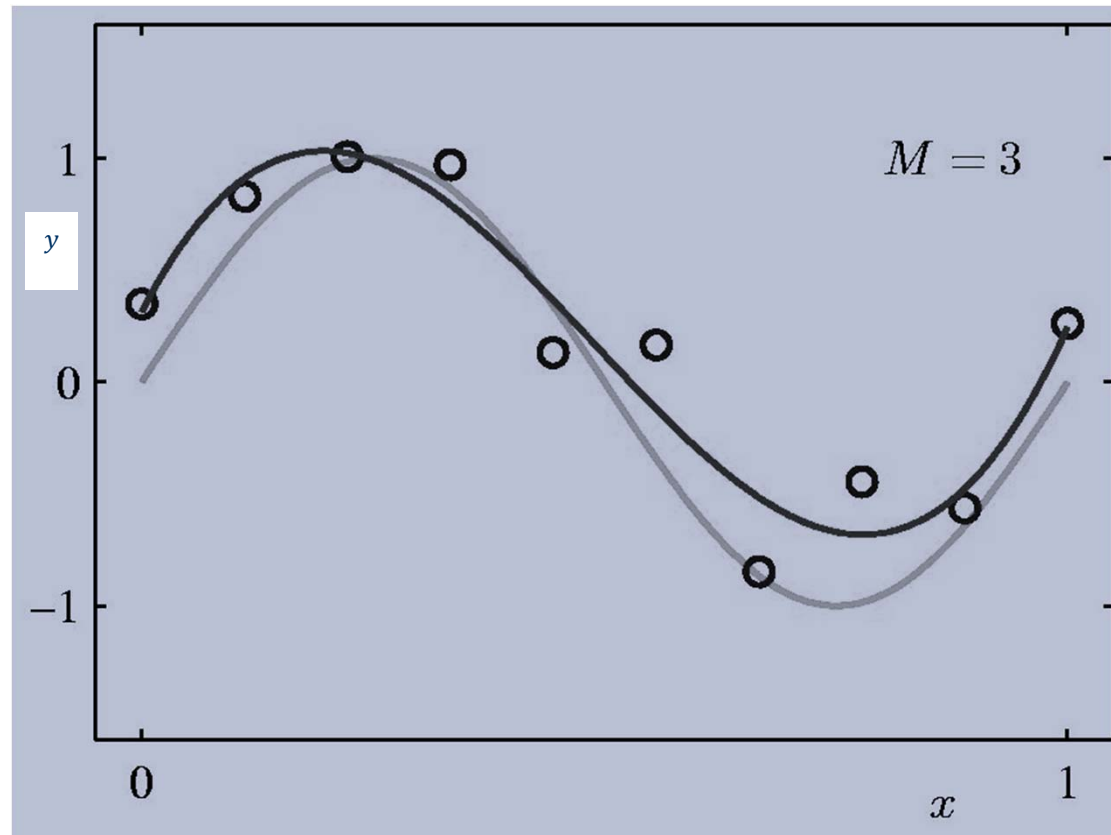


1st Order Polynomial



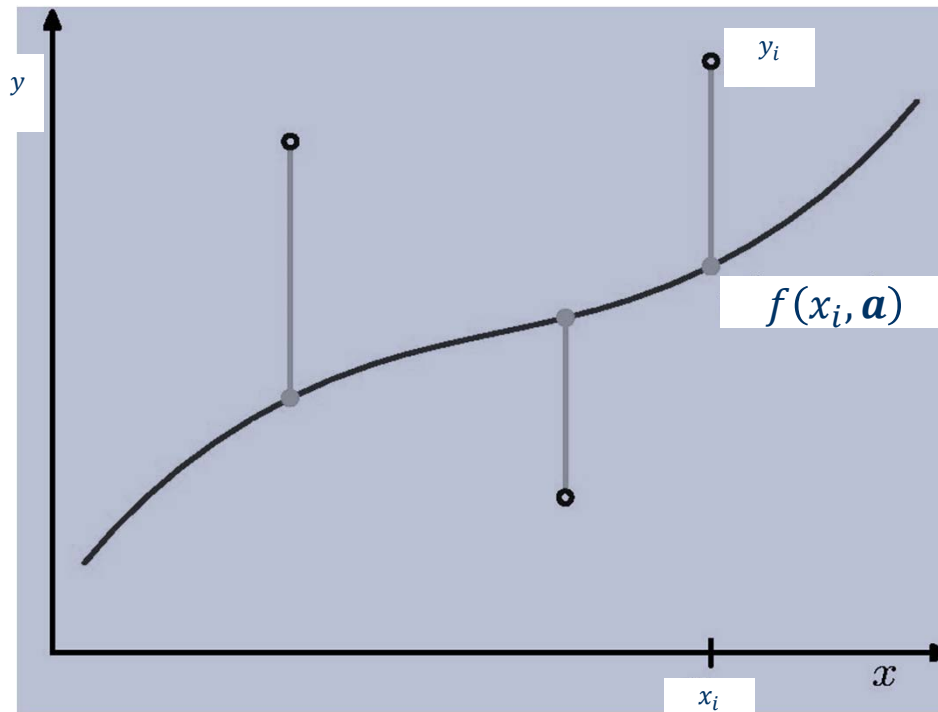


3rd Order Polynomial





Sum-of-Squares Error Function



- ▶ Training data:

- $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

- ▶ To learn f which $f(x) = y$

- ▶ Criterion function:

- $$\text{MSE}(\mathbf{a}) = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i, \mathbf{a}))^2$$



Polynomial Curve Fitting \rightarrow Linear Regression

$$f(x, \mathbf{a}) = a_0 + a_1x + a_2x^2 + \cdots + a_Mx^M = \sum_{j=1}^M a_Mx^M$$

- ▶ $\mathbf{x} = [1, x, x^2, \cdots, x^M]^T$
- ▶ $\mathbf{a} = [a_0, a_1, a_2, \cdots, a_M]^T$
- ▶ $f(x, \mathbf{a}) = \mathbf{a}^T \mathbf{x}$



Linear Regression Model

- ▶ Training data: (\mathbf{x}_i, y_i)
- ▶ $f(\mathbf{x}) = a_0 + \sum_{i=1}^p a_i x_i = a_0 + \mathbf{a}^T \mathbf{x}$
 - $\mathbf{a} = [a_1, \dots, a_p]^T$ and a_0 : unknown parameters or coefficients
 - \mathbf{x} : Feature vector, the outcome of **feature extraction**.
- ▶ Minimize the **mean-squared error** :

$$J_n = \frac{1}{n} \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2$$

- ▶ Minimize the **residual sum of squares**

$$RSS(f) = \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2$$



The MSE Criterion

$$J_n(\mathbf{a}) = \sum_{i=1}^n (y_i - \mathbf{a}^T \mathbf{x}_i)^2$$

- ▶ MSE Criterion: Minimize the sum of squared differences between $\mathbf{a}^T \mathbf{x}_i$ and y_i
- ▶ Using matrix notation for convenience

$$X = [\mathbf{x}_1, \dots, \mathbf{x}_n], \quad \mathbf{y} = [y_1, \dots, y_n]^T$$

$$J_n(\mathbf{a}) = (\mathbf{y} - X^T \mathbf{a})^T (\mathbf{y} - X^T \mathbf{a})$$

- ▶ How to optimize it (finds the optimal solution)?



Optimizing the MSE Criterion

- ▶ Computing the gradient gives:

$$\nabla J_n = -2X(\mathbf{y} - X^T \mathbf{a})$$

- ▶ Setting the gradient to zero,

$$XX^T \mathbf{a} = X\mathbf{y}$$

$$\mathbf{a} = (XX^T)^{-1} X\mathbf{y}$$

- ▶ Any problems?
- ▶ What is the rank of the matrix XX^T ?
- ▶ The solution for \mathbf{a} can be obtained uniquely if XX^T is non-singular.
- ▶ The fitted values at the training inputs are

$$\hat{\mathbf{y}} = X^T \mathbf{a} = X^T (XX^T)^{-1} X\mathbf{y}$$



Geometry of least-squares fitting

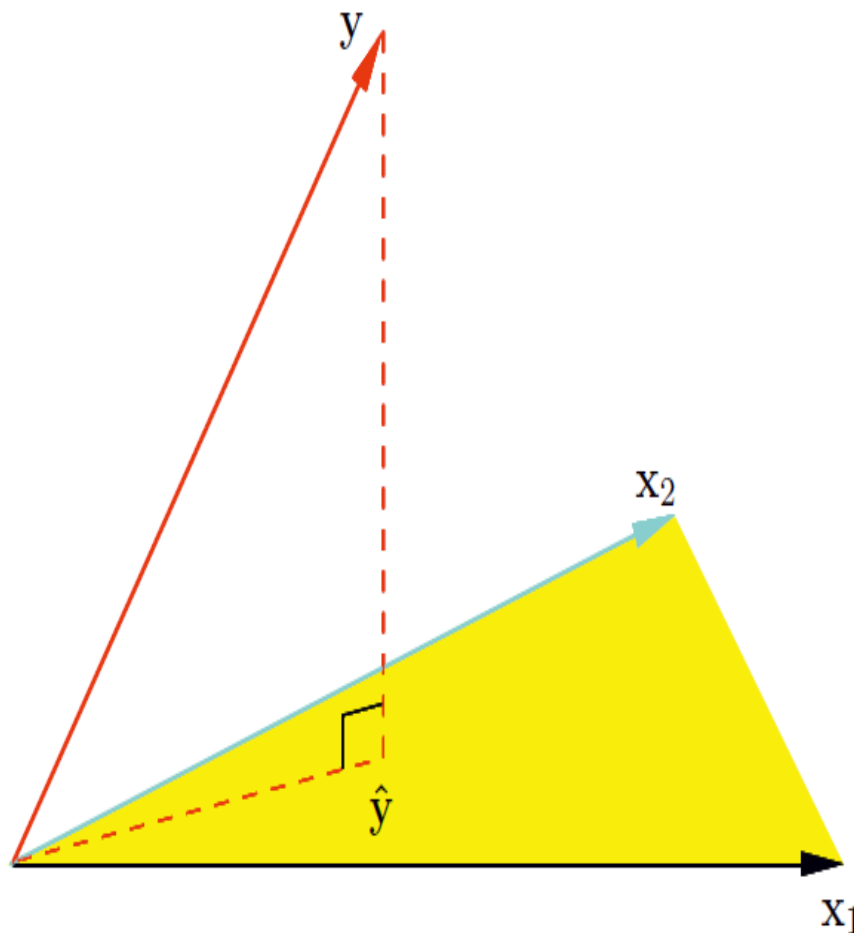
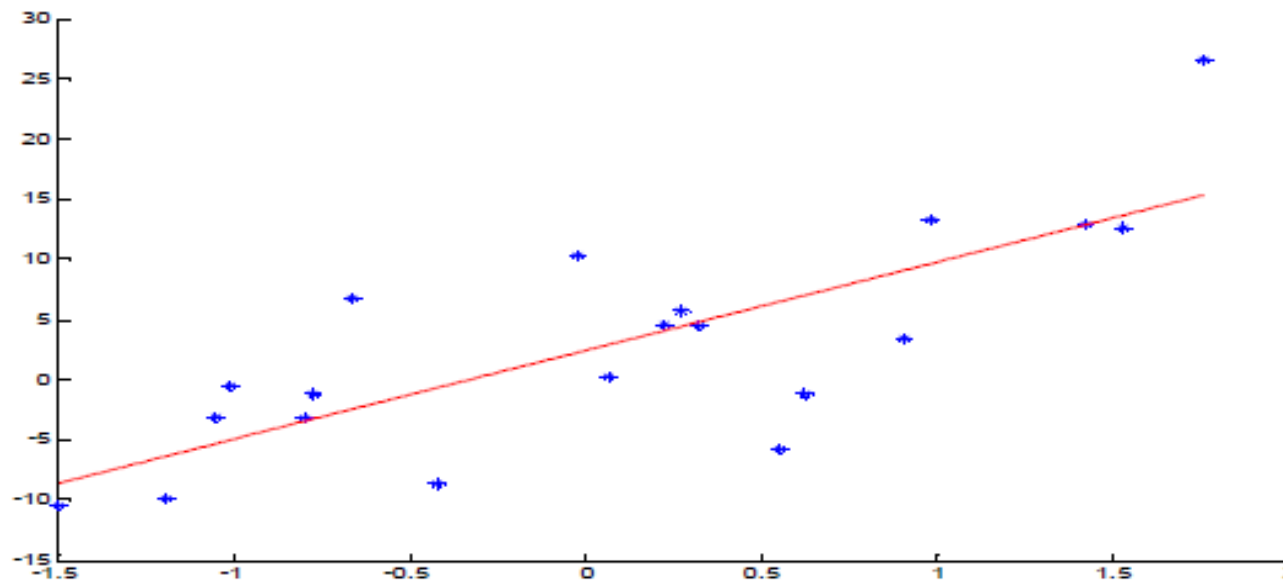


Figure 2: The N-dimensional geometry of least squares regression with two predictors. The outcome vector y is orthogonally projected onto the hyperplane spanned by the input vectors x_1 and x_2 . The projection \hat{y} represents the vector of the least squares predictions



Statistical model of regression

- ▶ A generative model: $y = f(\mathbf{x}, \mathbf{a}) + \epsilon$
- ▶ $f(\mathbf{x}, \mathbf{a})$ is a deterministic function
- ▶ ϵ is a random noise , it represents things we cannot capture with ,
e.g. $\epsilon \sim N(0, \sigma^2)$





Statistical model of regression

- ▶ A generative model: $y = f(\mathbf{x}, \mathbf{a}) + \epsilon$
- ▶ Assume: $\epsilon \sim N(0, \sigma^2)$
- ▶ $p(y|\mathbf{x}, \mathbf{a}, \sigma) = ?$

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(y-f(\mathbf{x},\mathbf{w}))^2}$$

- ▶ **Likelihood of predictions**
 - The probability of observing outputs \mathbf{y} in \mathbf{D} given \mathbf{a} , \mathbf{X} , and σ .



Maximum Likelihood Estimation

- ▶ Likelihood of predictions

$$L(\mathbf{D}, \mathbf{a}, \sigma) = \prod_{i=1}^n p(y_i | \mathbf{x}_i, \mathbf{a}, \sigma)$$

- ▶ Maximum likelihood estimation of parameters
 - Parameters maximizing the likelihood of predictions

$$\mathbf{a}^* = \operatorname{argmax} \prod_{i=1}^n p(y_i | \mathbf{x}_i, \mathbf{a}, \sigma)$$

- ▶ Log-likelihood



Maximum Likelihood Estimation

- Log-likelihood

$$l(\mathbf{D}, \mathbf{a}, \sigma) = \log(L(\mathbf{D}, \mathbf{a}, \sigma)) = \log \prod_{i=1}^n p(y_i | \mathbf{x}_i, \mathbf{a}, \sigma)$$

$$= \sum_{i=1}^n \log p(y_i | \mathbf{x}_i, \mathbf{a}, \sigma)$$

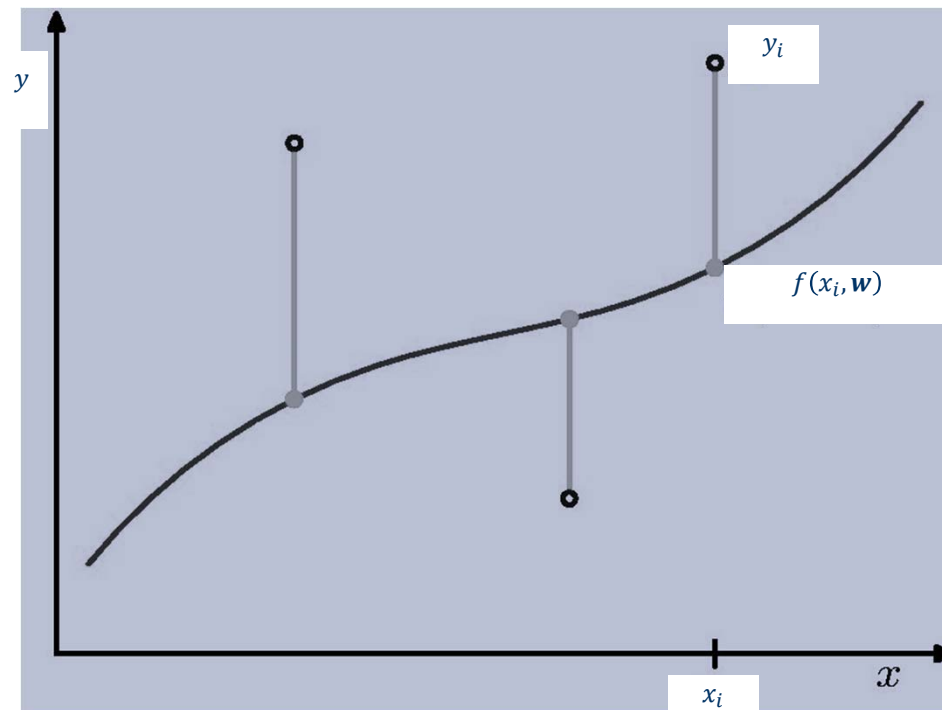
$$p(y_i | \mathbf{x}_i, \mathbf{a}, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(y - f(\mathbf{x}, \mathbf{a}))^2}$$

$$l(\mathbf{D}, \mathbf{a}, \sigma) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (y - f(\mathbf{x}, \mathbf{a}))^2 + c(\sigma)$$

$$RSS(f) = \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2$$



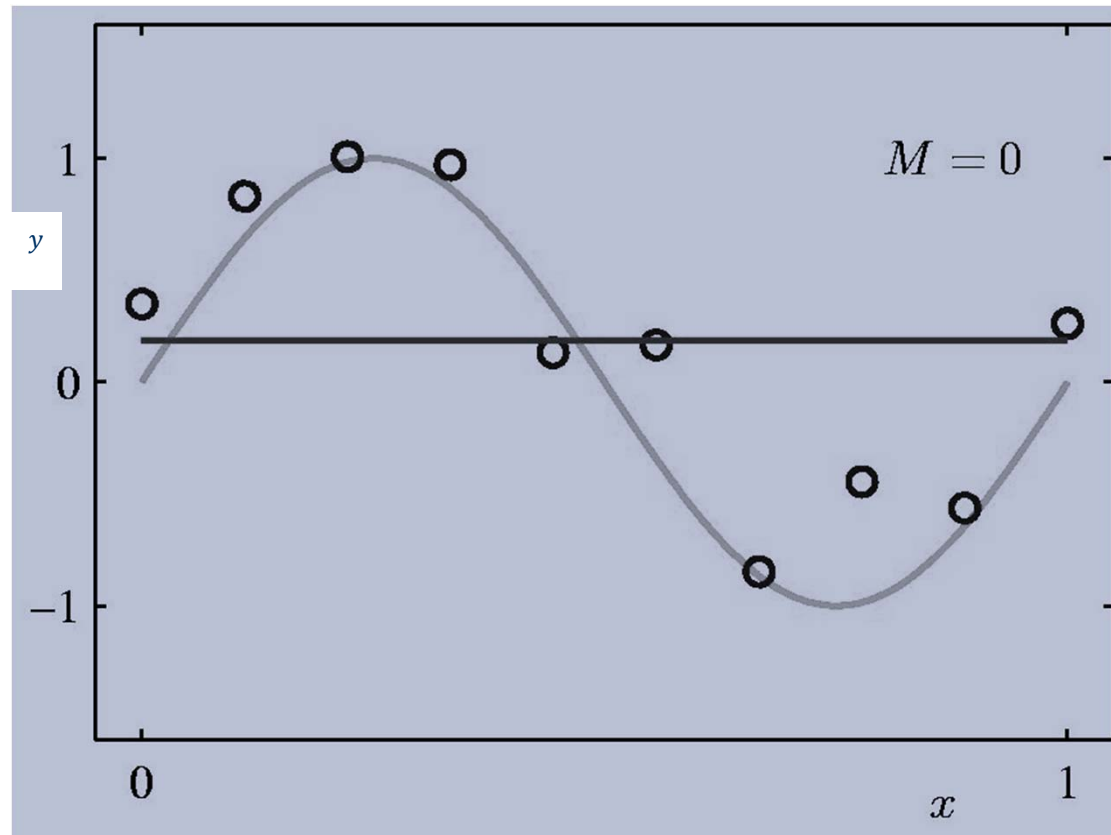
Sum-of-Squares Error Function



$$\text{MSE}(\mathbf{a}) = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i, \mathbf{a}))^2$$

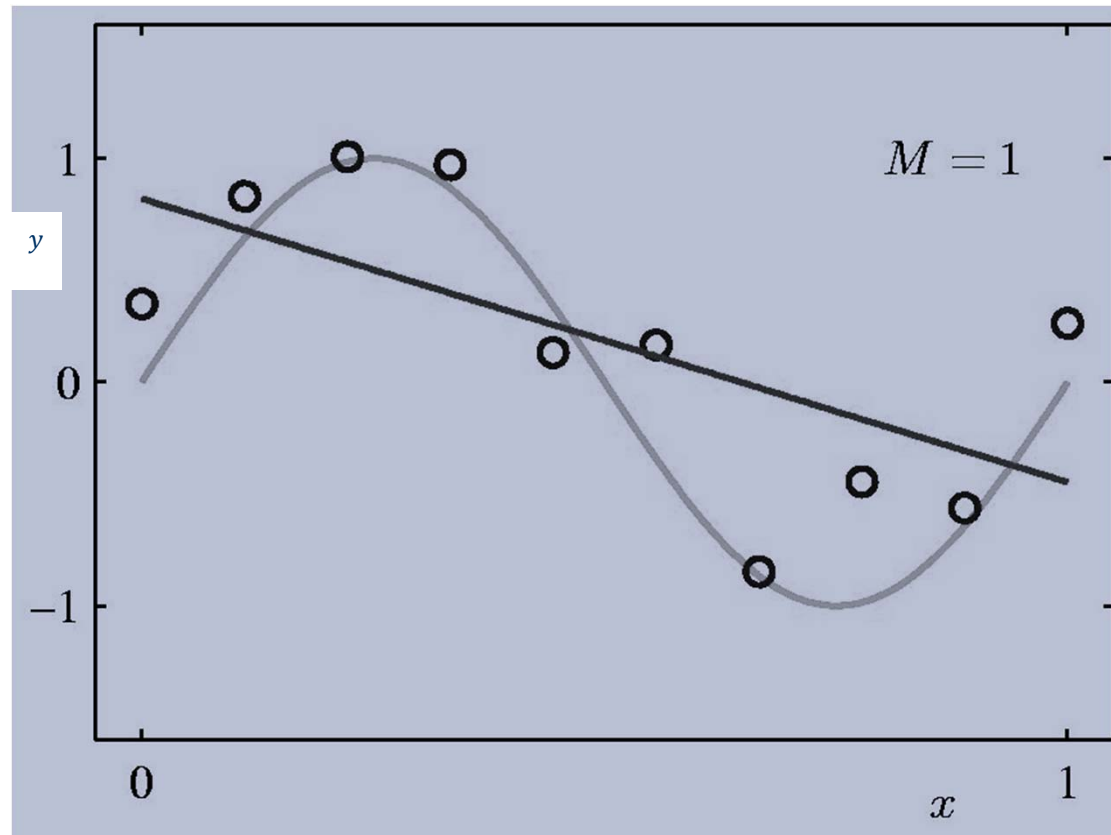


0th Order Polynomial



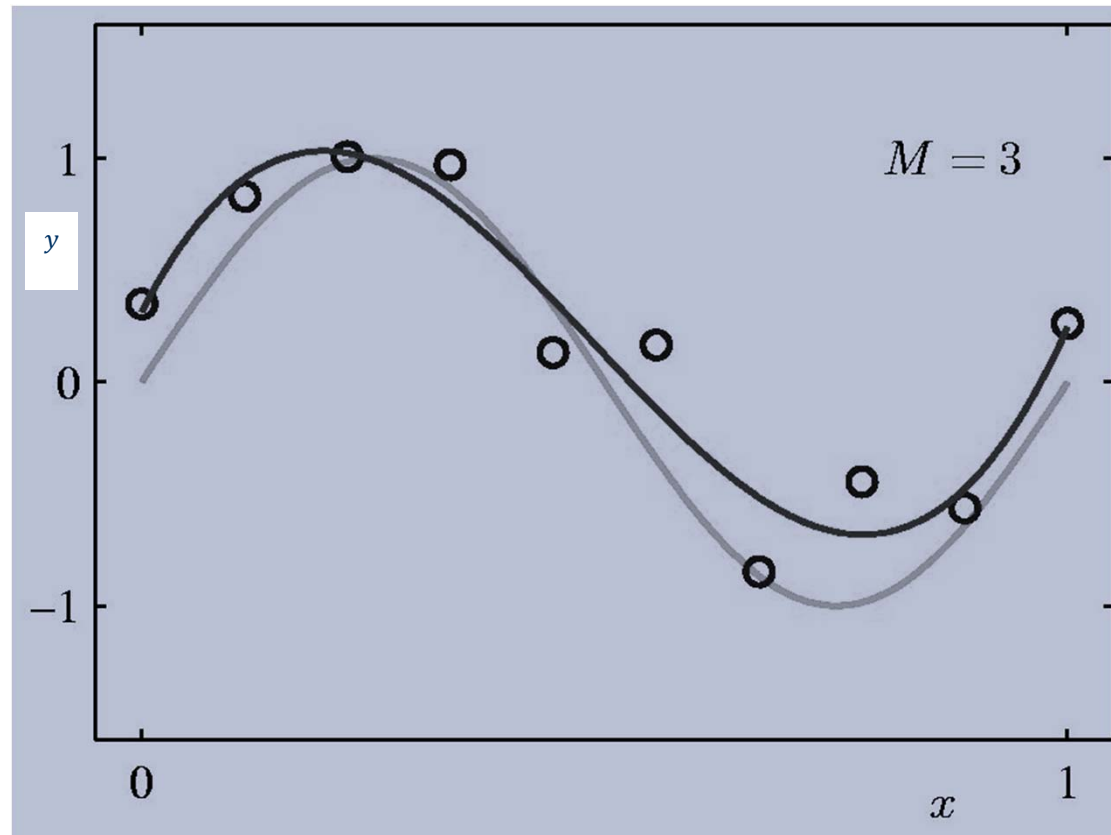


1st Order Polynomial



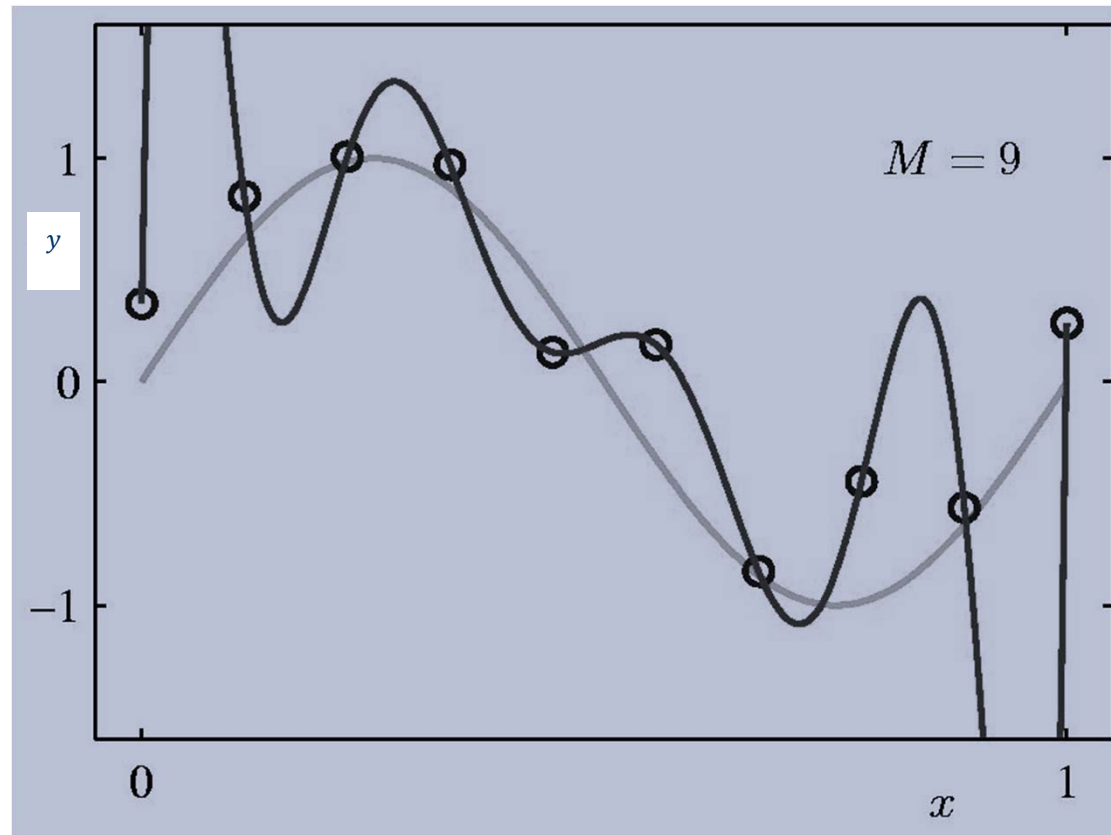


3rd Order Polynomial



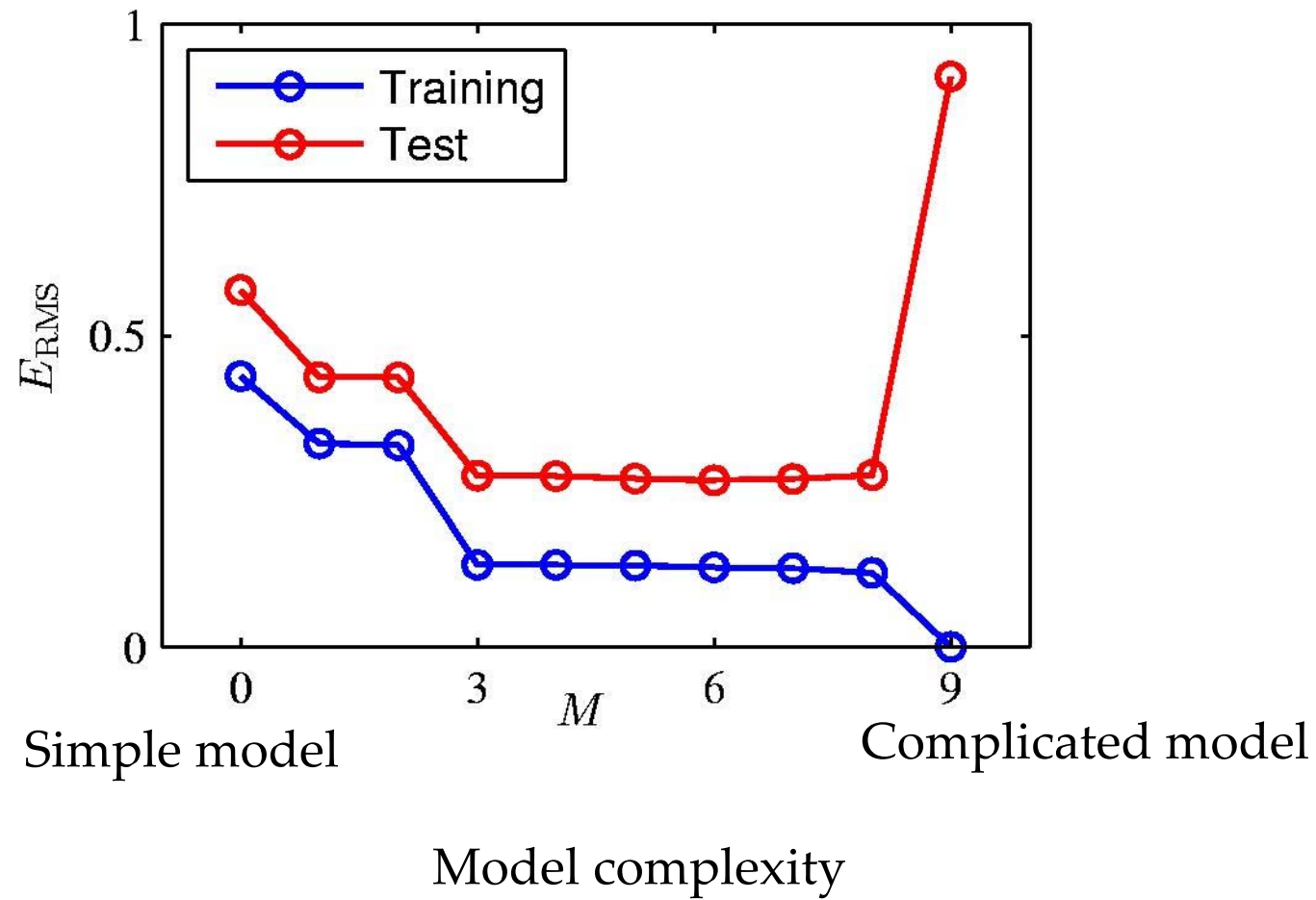


9th Order Polynomial





Over-fitting





Issues with MSE Criterion

$$\mathbf{a} = (XX^T)^{-1}X\mathbf{y}$$

- ▶ The solution for \mathbf{a} can be obtained uniquely if XX^T is non-singular.
- ▶ If XX^T is singular, overfitting

- ▶ Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

- ▶ Why?

$$y = f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} \quad \mathbf{w} = f'(\mathbf{x}) = \frac{dx}{dy}$$



Ridge Regression

- How to control the size of the coefficients in Regression?

$$\mathbf{a}^* = \operatorname{argmin} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \mathbf{a})^2 + \lambda \sum_{j=1}^p a_j^2$$

local smoothness

weight decay

- Equivalent formulation

$$\mathbf{a}^* = \operatorname{argmin} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \mathbf{a})^2$$

Subject to $\sum_{j=1}^p a_j^2 \leq t$

Lagrange multipliers



Ridge Regression

$$\mathbf{a}^* = \operatorname{argmin} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \mathbf{a})^2 + \lambda \sum_{j=1}^p a_j^2$$

- Matrix notations:

$$(\mathbf{y} - X^T \mathbf{a})^T (\mathbf{y} - X^T \mathbf{a}) + \lambda \mathbf{a}^T \mathbf{a}$$

- ▶ Computing the gradient gives:

$$-2X(\mathbf{y} - X^T \mathbf{a}) + 2\lambda \mathbf{a}$$

- ▶ Setting the gradient to zero,

$$(XX^T + \lambda I)\mathbf{a} = X\mathbf{y}$$

- ▶ The unique solution:

$$\mathbf{a}^* = (XX^T + \lambda I)^{-1} X\mathbf{y}$$

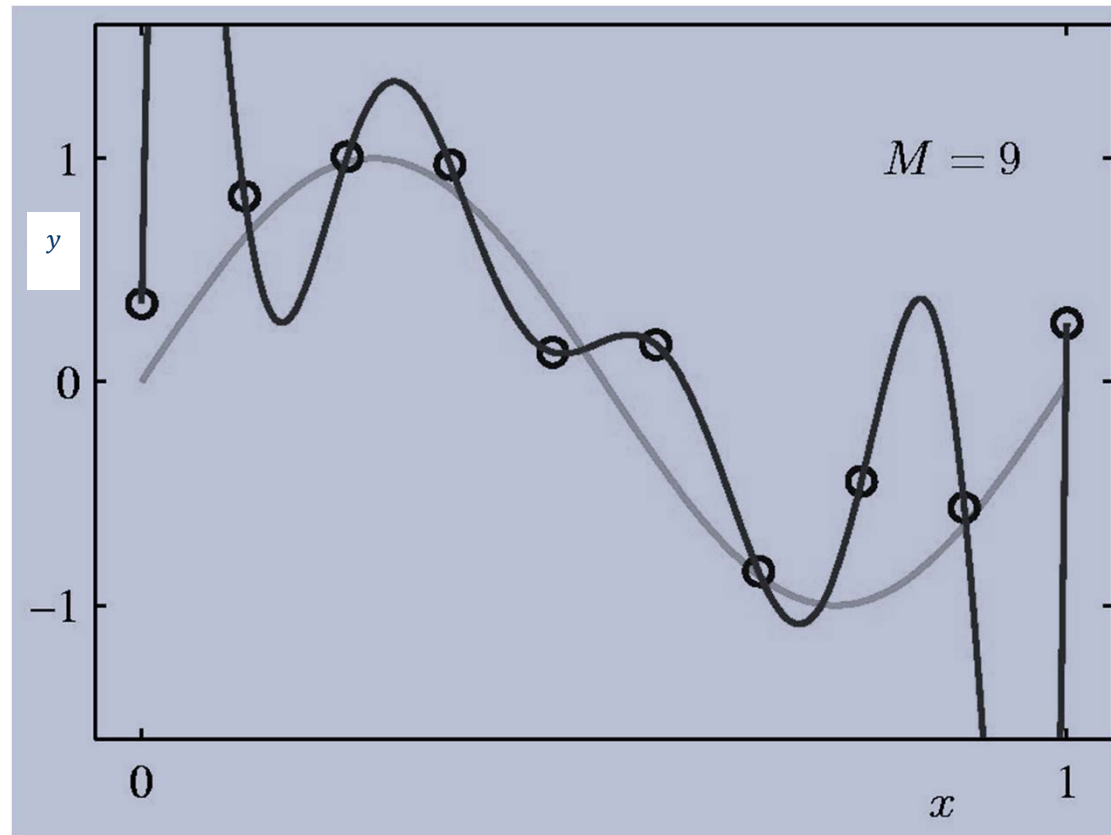


Polynomial Coefficients

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

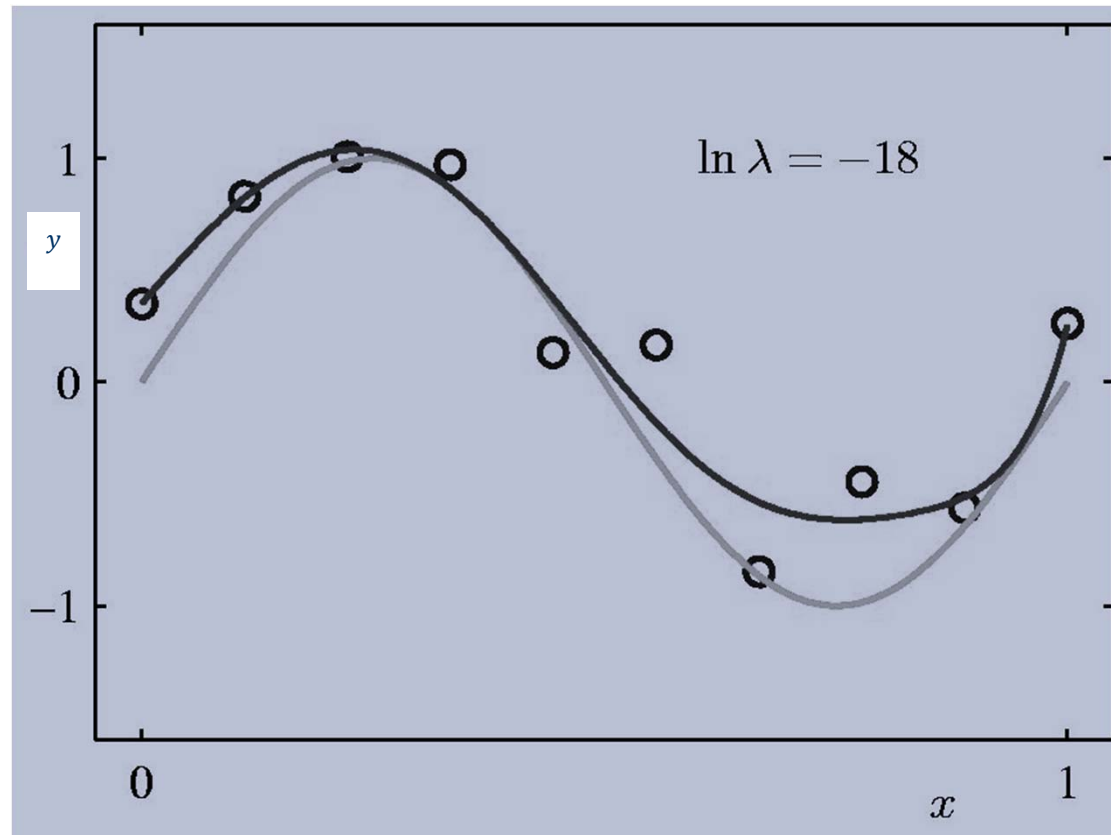


9th Order Polynomial



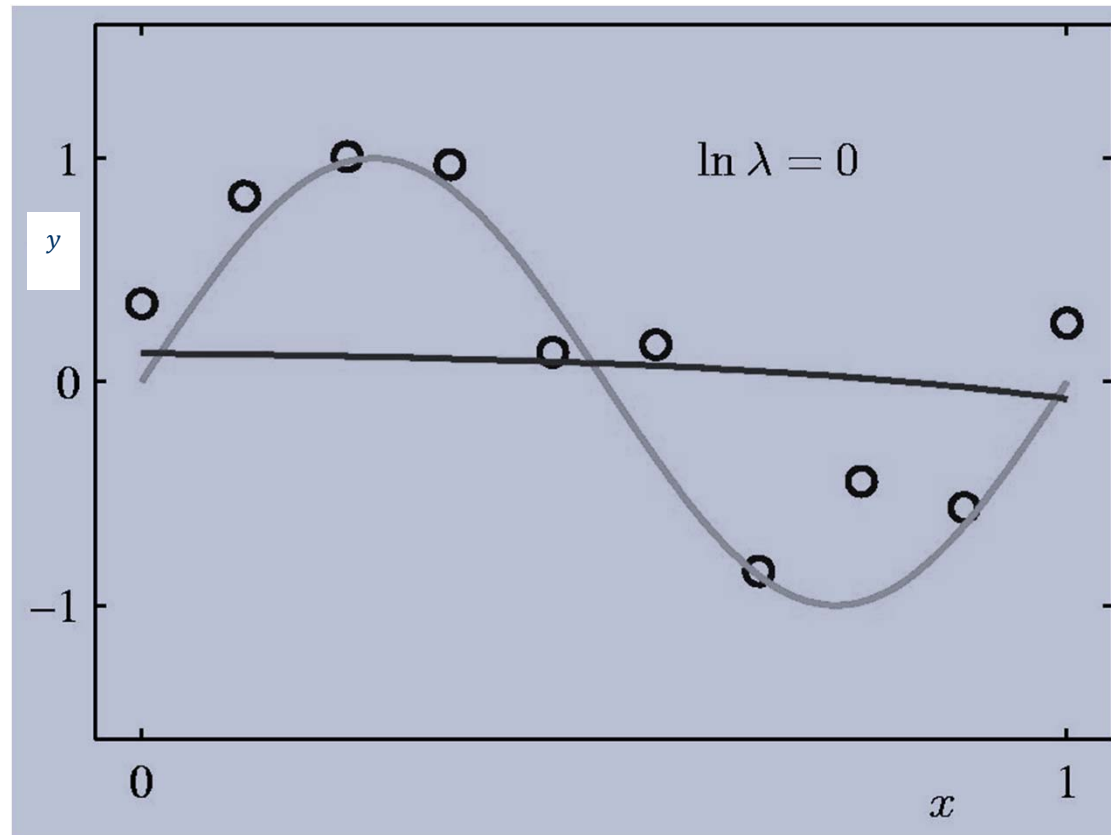


Regularization ($\ln \lambda = -18$)



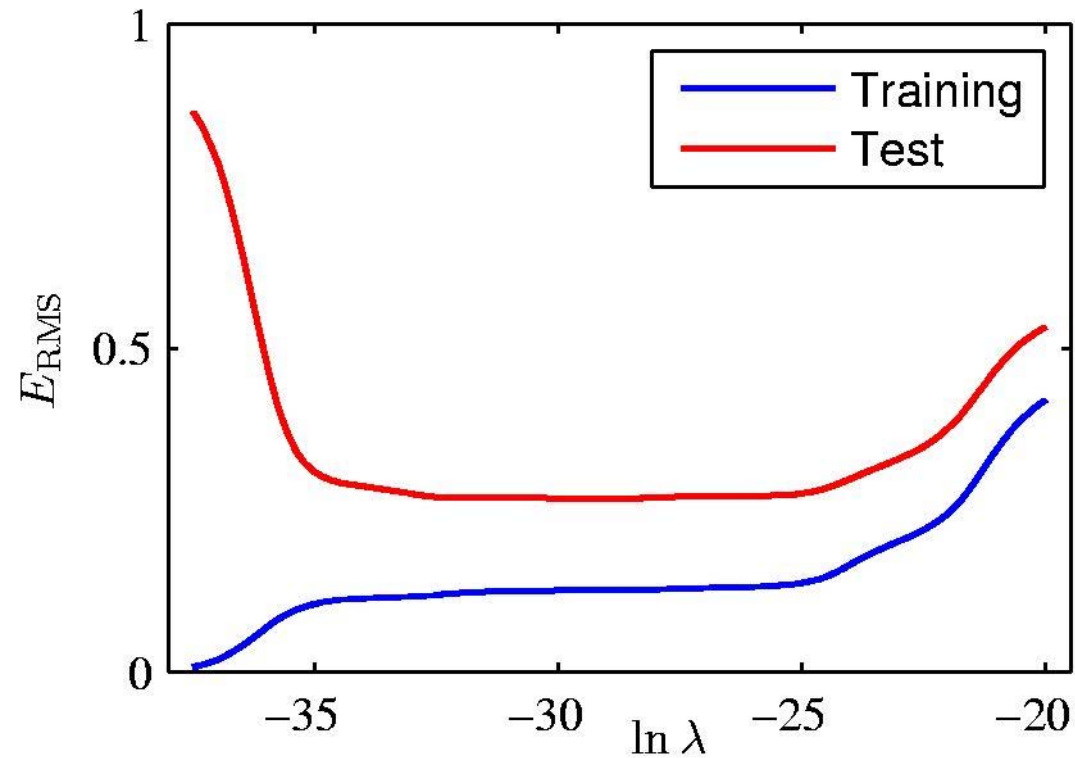


Regularization ($\ln \lambda = 0$)





Regularization



Complicated model

Simple model

Model complexity



Maximum Likelihood Estimation

► A generative model: $y = f(\mathbf{x}, \mathbf{a}) + \epsilon$

► Assume: $\epsilon \sim N(0, \sigma^2)$

► $p(y|\mathbf{x}, \mathbf{a}, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(y-f(\mathbf{x},\mathbf{a}))^2}$

$$\mathbf{a}^* = \operatorname{argmax} L(\mathbf{D}, \mathbf{a}, \sigma) = \operatorname{argmax} \prod_{i=1}^n p(y_i|\mathbf{x}_i, \mathbf{a}, \sigma)$$

► $l(\mathbf{D}, \mathbf{a}, \sigma) = \log(L(\mathbf{D}, \mathbf{a}, \sigma)) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (y - f(\mathbf{x}, \mathbf{a}))^2 + c(\sigma)$



Bayesian Linear Regression

► Bayes rule

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

$$P(\mathbf{a}|y, \mathbf{x}, \sigma) = \frac{P(y|\mathbf{a}, \mathbf{x}, \sigma)P(\mathbf{a}|\mathbf{x}, \sigma)}{P(y|\mathbf{x}, \sigma)}$$

$$\text{posterior} \quad \boxed{P(\mathbf{a}|\mathcal{D})} = \frac{\overbrace{P(\mathcal{D}|\mathbf{a})}^{\text{likelihood}} \overbrace{P(\mathbf{a})}^{\text{prior}}}{P(\mathcal{D})}$$

$$\text{Posterior} \propto \text{likelihood} \times \text{prior}$$



Bayesian Linear Regression

Posterior \propto likelihood \times prior

- A common choice for the prior is

$$\begin{aligned} p(\mathbf{a}) &= \mathcal{N}(\mathbf{a} | \mathbf{0}, \lambda^{-1} \mathbf{I}) \\ &= \frac{1}{(2\pi)^{\frac{q}{2}} |\lambda^{-1} \mathbf{I}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{a}-\mathbf{0})^T (\lambda^{-1} \mathbf{I})^{-1} (\mathbf{a}-\mathbf{0})} \end{aligned}$$

$$\ln(p(\mathbf{a})) = -\frac{\lambda}{2} \mathbf{a}^T \mathbf{a} + c$$

$$l(\mathbf{D}, \mathbf{a}, \sigma) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (y - f(\mathbf{x}, \mathbf{a}))^2 + c(\sigma)$$



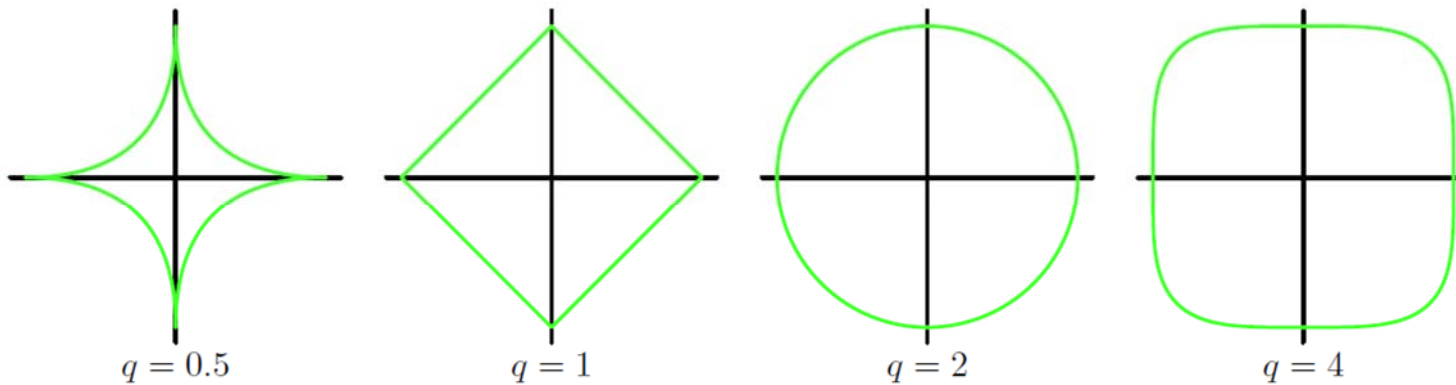
A more general regularizer

► Ridge Regression

$$\hat{\beta} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - x_i^T \beta)^2$$

subject to $\sum_{j=1}^p \beta_j^2 \leq t$

subject to $\sum_{j=1}^p |\beta_j|^q \leq t$





LASSO

- Least Absolute Selection and Shrinkage Operator

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2$$

subject to $\sum_{j=1}^p |\beta_j| \leq t$

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_1$$

- Sparse model

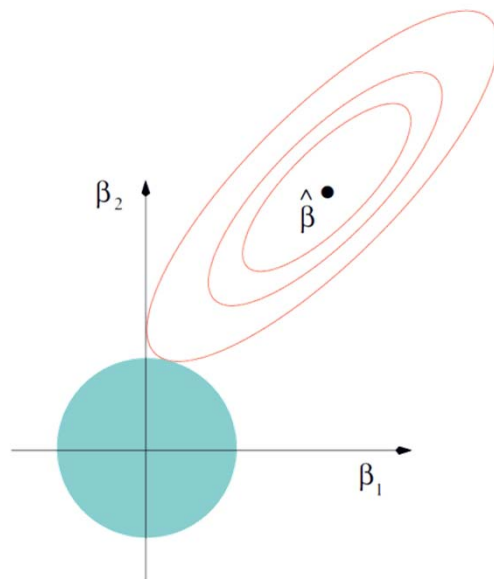


LASSO: Sparse Model

- ▶ Ridge regression VS. LASSO
 - Why LASSO \rightarrow Sparse model ?

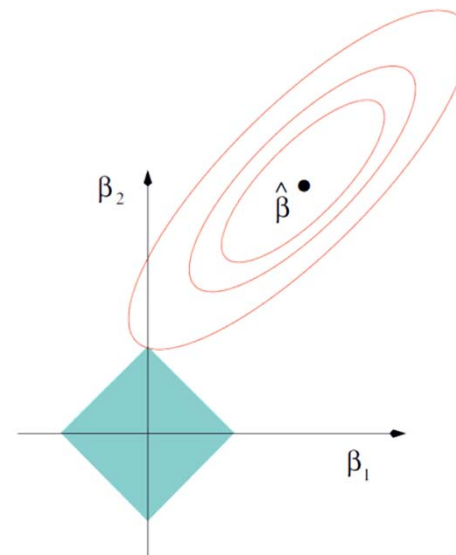
$$\hat{\beta} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2$$

subject to $\sum_{j=1}^p \beta_j^2 \leq t$



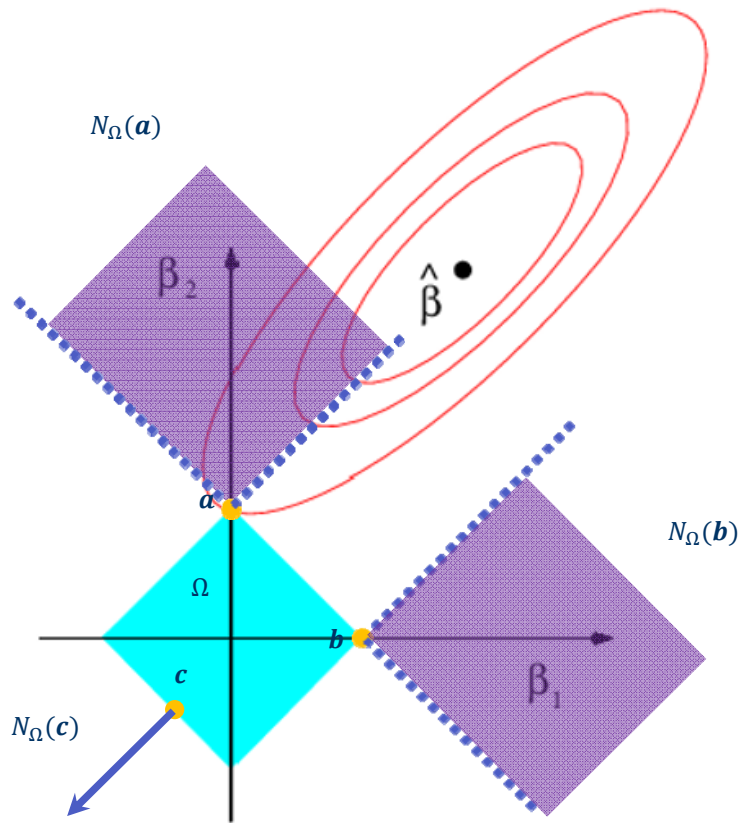
$$\hat{\beta} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2$$

subject to $\sum_{j=1}^p |\beta_j| \leq t$





LASSO: Sparse Model



- ▶ The normal cones to the feasible set at the corner points, such as \mathbf{a} and \mathbf{b} , contain infinitely many rays, while they reduce to a singleton (only contain a single ray) at the other boundary points.
- ▶ The first order (necessary and sufficient) condition concludes that: a feasible point becomes the optimum if and only if the opposite gradient direction of the objective function falls inside the normal cone to the feasible set at that point.
- ▶ Thus, the optimum will more likely fall at the points with “larger” normal cones.
- ▶ This also explains why non-convex regularizers usually induce sparser models than the convex regularizers.



LASSO Solution

$$\frac{1}{n} \sum_i y_i = 0, \quad \frac{1}{n} \sum_i \mathbf{x}_i = \mathbf{0}, \quad \frac{1}{n} \sum_i x_{ij}^2 = 1$$

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_1$$

- Convex optimization
- Coordinate descent
- Single predictor (feature) setting

$$\hat{\beta} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - z_i \beta)^2 + \lambda |\beta|$$

$$\hat{\beta} = \frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle - \lambda \quad \text{If } \frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle > \lambda$$

$$\hat{\beta} = 0 \quad \text{If } \left| \frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle \right| \leq \lambda$$

$$\hat{\beta} = \frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle + \lambda \quad \text{If } \frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle < -\lambda$$



LASSO Solution

► Single predictor (feature) setting

$$\hat{\beta} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - z_i \beta)^2 + \lambda |\beta| \quad \frac{1}{n} \sum_i y_i = 0, \quad \frac{1}{n} \sum_i z_i = 0, \quad \frac{1}{n} \sum_i z_i^2 = 1$$

$$f(\beta) = \frac{1}{2n} \sum_{i=1}^n (y_i - z_i \beta)^2 + \lambda |\beta| = \frac{1}{2n} (\mathbf{y} - \beta \mathbf{z})^T (\mathbf{y} - \beta \mathbf{z}) + \lambda |\beta|$$

$$f(\beta) = \frac{1}{2n} \mathbf{z}^T \mathbf{z} \beta^2 - \frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle \beta + \lambda |\beta| + \frac{1}{2n} \mathbf{y}^T \mathbf{y}$$

$$f(\beta) = \frac{1}{2} \beta^2 - \frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle \beta + \lambda |\beta|$$

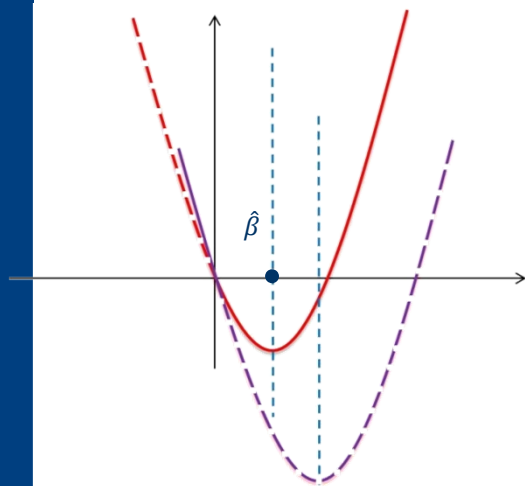
$$f(\beta) = \begin{cases} \frac{1}{2} \beta^2 - \left(\frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle - \lambda \right) \beta, & \beta \geq 0 \\ \frac{1}{2} \beta^2 - \left(\frac{1}{n} \langle \mathbf{z}, \mathbf{y} \rangle + \lambda \right) \beta, & \beta < 0 \end{cases}$$



LASSO Solution

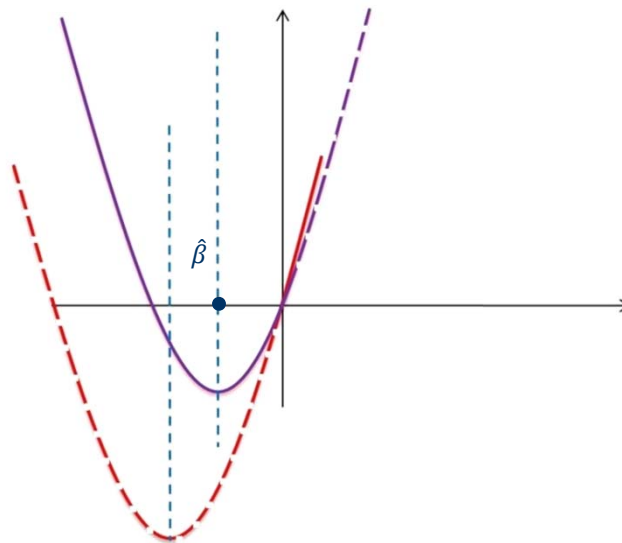
$$f(\beta) = \begin{cases} \frac{1}{2}\beta^2 - \left(\frac{1}{n}\langle \mathbf{z}, \mathbf{y} \rangle - \lambda\right)\beta, & \beta \geq 0 \\ \frac{1}{2}\beta^2 - \left(\frac{1}{n}\langle \mathbf{z}, \mathbf{y} \rangle + \lambda\right)\beta, & \beta < 0 \end{cases}$$

$$\frac{1}{n}\langle \mathbf{z}, \mathbf{y} \rangle > \lambda$$



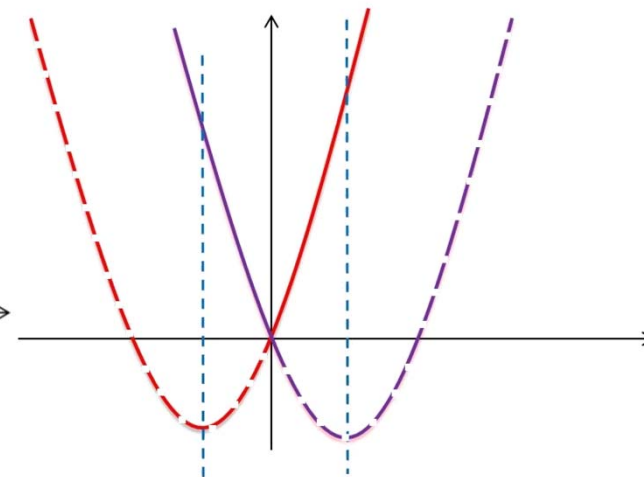
$$\hat{\beta} = \frac{1}{n}\langle \mathbf{z}, \mathbf{y} \rangle - \lambda$$

$$\frac{1}{n}\langle \mathbf{z}, \mathbf{y} \rangle < -\lambda$$



$$\hat{\beta} = \frac{1}{n}\langle \mathbf{z}, \mathbf{y} \rangle + \lambda$$

$$\left| \frac{1}{n}\langle \mathbf{z}, \mathbf{y} \rangle \right| < \lambda$$



$$\hat{\beta} = 0$$



LASSO Solution

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin} \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_1$$

- ▶ Convex optimization
- ▶ Coordinate descent
- ▶ Single predictor (feature) setting
- ▶ Multiple predictors (features)
 - Cyclic Coordinate descent
- ▶ LARs



Model Assessment and Selection

- ▶ The generalization performance of a learning method
- ▶ Model selection:
 - estimating the performance of different models in order to choose the best one.
- ▶ Model assessment:
 - having chosen a final model, estimating its prediction error (generalization error) on new data.



Bias & Variance Decomposition



The Supervised Learning Problem

- ▶ **Given** example pairs $[\mathbf{x}_i, y_i]$
- ▶ **Learn** a function $f(\mathbf{x})$, such as $f(\mathbf{x})=y$
- ▶ **Loss:** $L(y, f(\mathbf{x}))$
- ▶ Expected Loss:

$$E(L) = \iint L(y, f(\mathbf{x})) p(\mathbf{x}, y) d\mathbf{x} dy$$

- ▶ Squared loss: $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$
- ▶ Expected Prediction Error:

$$EPE(f) = \iint (y - f(\mathbf{x}))^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

$$f^*(\mathbf{x}) = ?$$



$$\text{EPE}(f) = \iint (y - f(\mathbf{x}))^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

- Squared loss

$$\begin{aligned} (y - f(\mathbf{x}))^2 &= (y - E(y|\mathbf{x}) + E(y|\mathbf{x}) - f(\mathbf{x}))^2 \\ &= \underbrace{(y - E(y|\mathbf{x}))^2}_{\text{Expected Prediction Error}} + \underbrace{(E(y|\mathbf{x}) - f(\mathbf{x}))^2}_{\text{Bias}} + 2(y - E(y|\mathbf{x}))(E(y|\mathbf{x}) - f(\mathbf{x})) \end{aligned}$$

- Expected Prediction Error:

$$\text{EPE}(f) = \int (f(\mathbf{x}) - E(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \int \text{var}(y|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

- The first term:

- $f^*(\mathbf{x}) = E(y|\mathbf{x})$

- The second term:



In Reality

- ▶ **Given** training set D , contains n example pairs $[\mathbf{x}_i, y_i]$
- ▶ **Learn** a function $f(\mathbf{x})$, such as $f(\mathbf{x})=y$
- ▶ Expected Prediction Error:

$$\text{EPE}(f) = \iint (y - f(\mathbf{x}))^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

$$f(\mathbf{x}) \rightarrow f(\mathbf{x}; D)$$
$$E_D(f(\mathbf{x}; D))$$



In Reality

- Expected Prediction Error:

$$\text{EPE}(f) = \int (f(\mathbf{x}) - E(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \int \text{var}(y|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$= \int (f(\mathbf{x}; D) - E(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \int \text{var}(y|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$[f(\mathbf{x}; D) - E_D(f(\mathbf{x}; D)) + E_D(f(\mathbf{x}; D)) - E(y|\mathbf{x})]^2$$

$$(f(\mathbf{x}; D) - E_D(f(\mathbf{x}; D)))^2 + (E_D(f(\mathbf{x}; D)) - E(y|\mathbf{x}))^2 + 2(f(\mathbf{x}; D) - E_D(f(\mathbf{x}; D)))(E_D(f(\mathbf{x}; D)) - E(y|\mathbf{x}))$$

$$E_D \left\{ (f(\mathbf{x}; D) - E(y|\mathbf{x}))^2 \right\}$$

$$\underbrace{E_D \left\{ [f(\mathbf{x}; D) - E_D(f(\mathbf{x}; D))]^2 \right\}}_{\text{Variance}} + \underbrace{\{E_D(f(\mathbf{x}; D)) - E(y|\mathbf{x})\}^2}_{(\text{Bias})^2}$$

Variance

(Bias)²



Bias-variance Decomposition

- ▶ $EPE(f) = \iint (y - f(x))^2 p(x, y) dx dy$

- ▶ Expected prediction error (expected loss) =

(bias)² + variance + noise

- ▶ (bias)²:

$$\int \{E_D(f(x; D)) - E(y|x)\}^2 p(x) dx$$

- ▶ variance:

$$\int E_D \{ [f(x; D) - E_D(f(x; D))]^2 \} p(x) dx$$

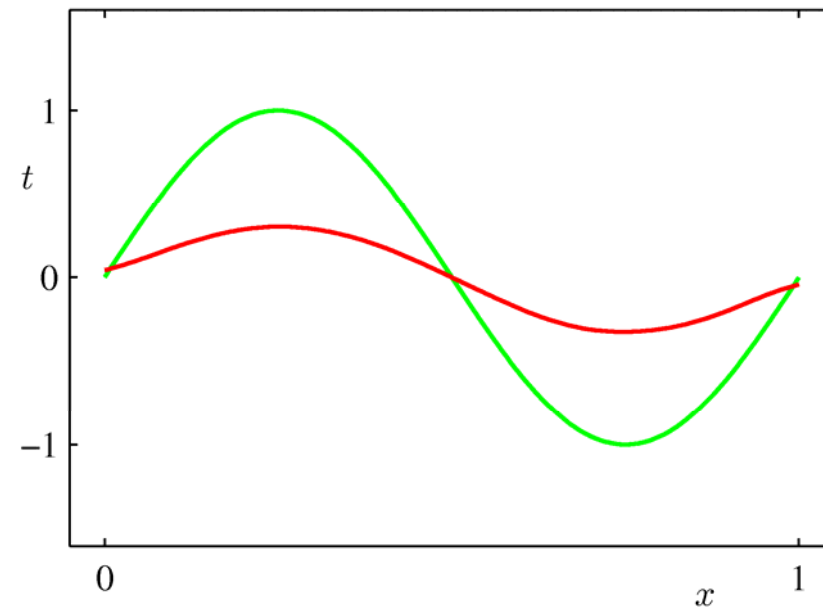
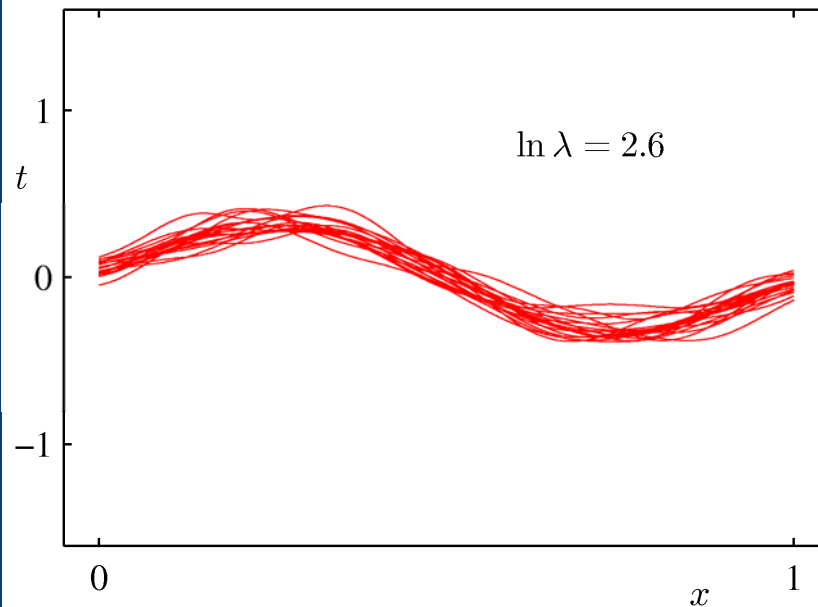
- ▶ noise:

$$\int \text{var}(y|x) p(x) dx$$



The Bias-Variance Decomposition

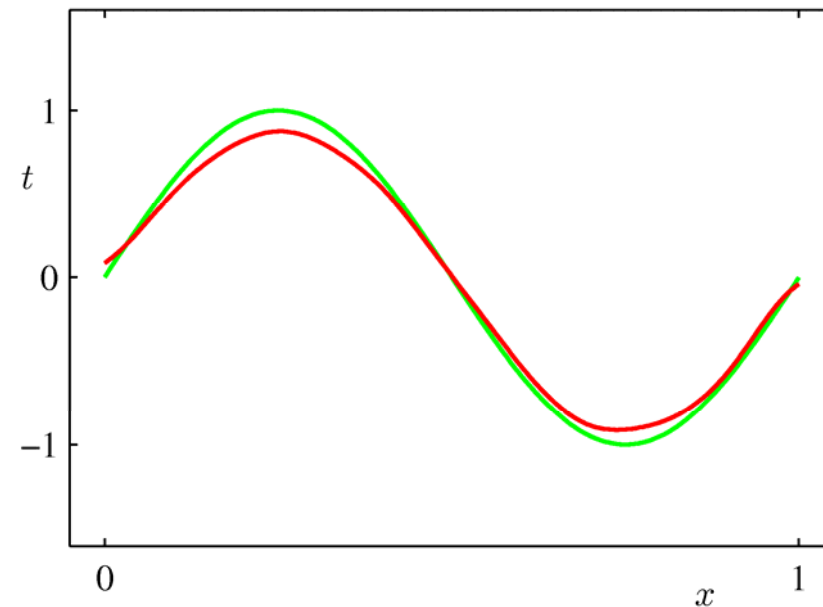
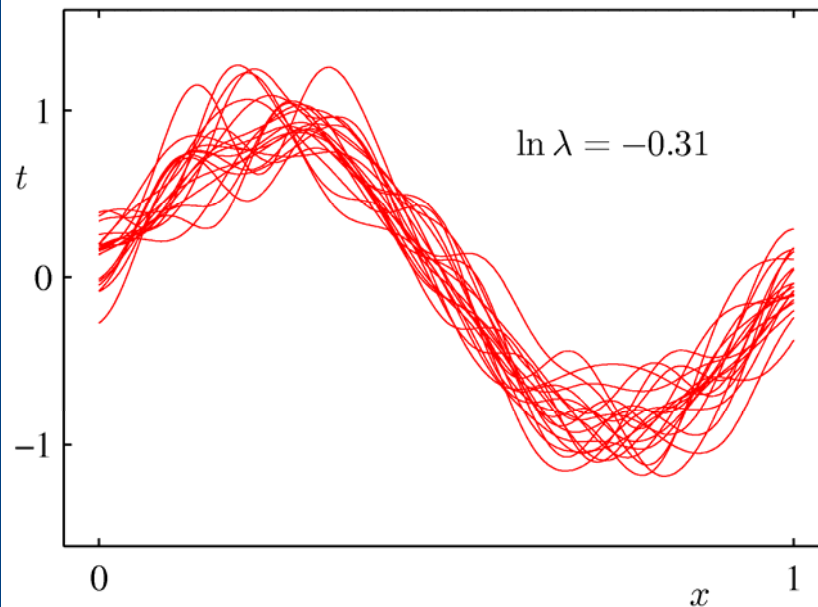
- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ .





The Bias-Variance Decomposition

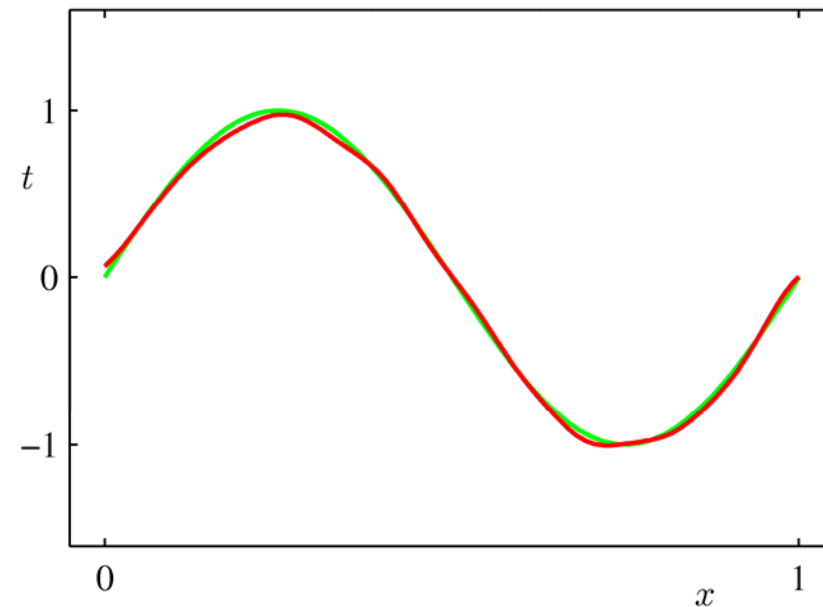
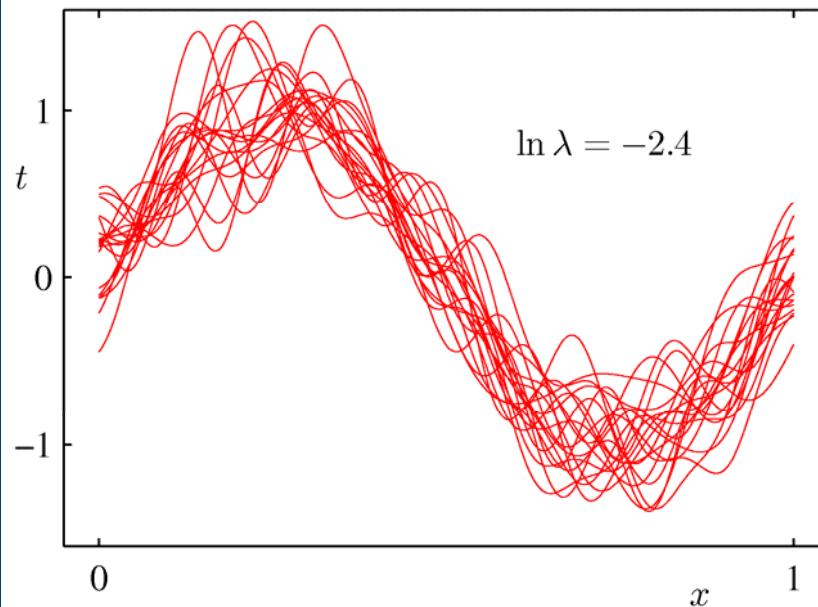
- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ .





The Bias-Variance Decomposition

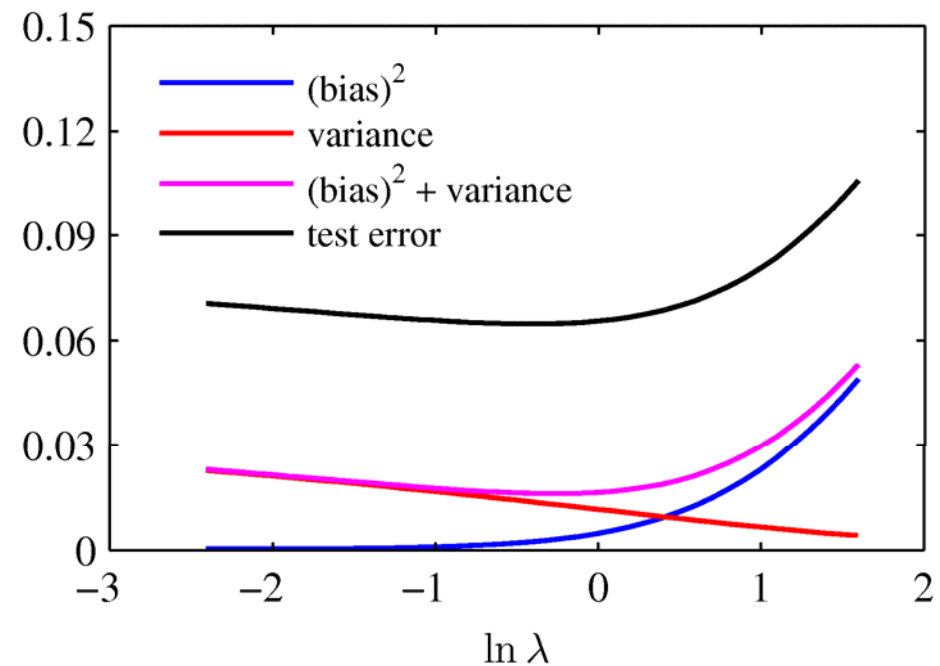
- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ .





The Bias-Variance Trade-off

- ▶ Over-regularized model (large λ) \rightarrow high bias
- ▶ Under-regularized model (small λ) \rightarrow high variance.





Cross-Validation

1	2	3	4	5
Train	Train	Validation	Train	Train

- ▶ K-Fold Cross-Validation
- ▶ leave-one-out cross-validation