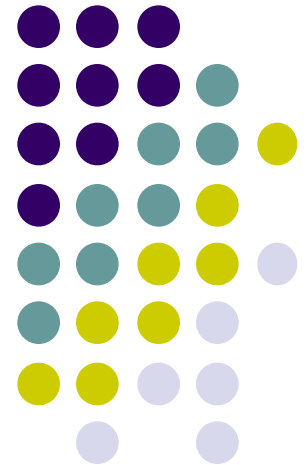


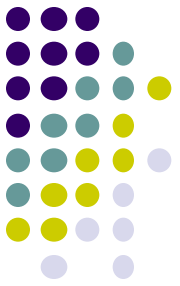
# Point Estimation

---

Hongxin Zhang  
zhx@cad.zju.edu.cn

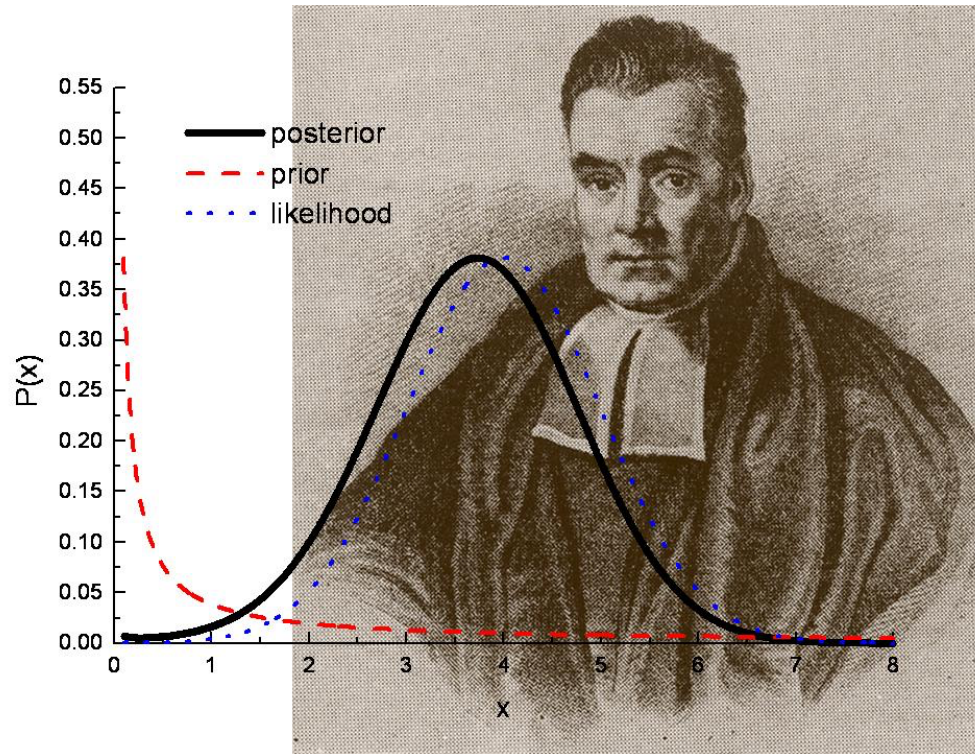
State Key Lab of CAD&CG, ZJU  
2019-02-26





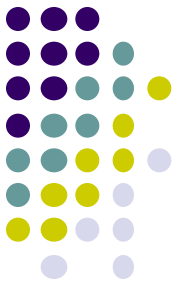
# What you need to know

- Point estimation: (点估计)
  - Maximal Likelihood Estimation (MLE)
  - Bayesian learning
  - Maximize **A** Posterior (MAP)
- Gaussian estimation
- Regression (回归)
  - Basis function = features
  - Optimizing sum squared error
  - Relationship between regression and Gaussians
- Bias-Variance trade-off



点估计是非常重要的数据计算技术，需要在入门阶段重点掌握

# Point estimation



# Your first consulting job

- An IT billionaire from USA asks you a question:
  - B: I have thumbtack, if I flip it, what's the probability it will fall with the nail up?
  - Y: Please flip it a few times ...



- Y: The probability is  $3/5$
- B: Why???
- Y: Because...



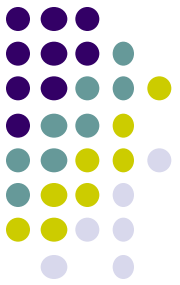
# Binomial Distribution

- $P(\text{Heads}) = \theta, P(\text{Tails}) = 1 - \theta \quad D = \{T, H, H, T, T\}$

$$P(D \mid \theta) = (1 - \theta)\theta\theta(1 - \theta)(1 - \theta)$$

- Flips are i.i.d. (Independent Identically distributed)
  - Independent events
  - Identically distributed according to Binomial distribution
- Sequence D of  $\alpha_H$  Heads and  $\alpha_T$  Tails

$$P(D \mid \theta) = \theta^{\alpha_H} (1 - \theta)^{\alpha_T}$$



# Maximum Likelihood Estimation

- **Data:** Observed set  $D$  of  $\alpha_H$  Heads and  $\alpha_T$  Tails
- **Hypothesis:** Binomial distribution
- **Learning  $\theta$  is an optimization problem**
  - What's the objective function?

$$D = \{T, H, H, T, T\}$$

- **MLE:** Choose  $\theta$  that maximizes the probability of observed data:

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} P(D | \theta) \\ &= \arg \max_{\theta} \ln P(D | \theta) = \dots\end{aligned}$$

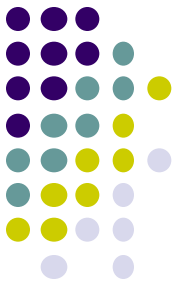
# Maximum Likelihood Estimation (cont.)



$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} P(D | \theta) \\ &= \arg \max_{\theta} \ln(\theta^{\alpha_H} (1 - \theta)^{\alpha_T}) \\ &= \arg \max_{\theta} (\alpha_H \ln \theta + \alpha_T \ln(1 - \theta))\end{aligned}$$

- Set derivative to zero:

$$\boxed{\frac{d}{d\theta} \ln P(D | \theta) = 0} \quad \hat{\theta} = \frac{\alpha_H}{\alpha_H + \alpha_T} = \frac{2}{2 + 3}$$



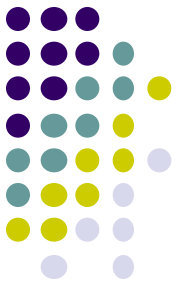
# How many flips do I need?

$$\hat{\theta} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

- B: I flipped 2 heads and 3 tails.
- Y:  $1 - \theta = 3/5$ , I can prove it!
- B: What if I flipped 20 heads and 30 tails?
- Y: Same answer, I can prove it!
- B: What's better?
- Y: Humm... The more the merrier???
- B: Is this why I am paying you the big bucks???



# Simple bound (based on Hoeffding's inequality)



- For  $N = \alpha_H + \alpha_T$  and  $\hat{\theta} = \frac{\alpha_T}{\alpha_H + \alpha_T}$

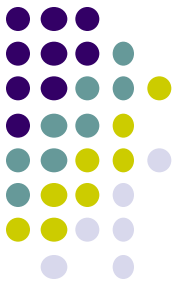
<http://omega.albany.edu:8008/machine-learning-dir/notes-dir/vc1/vc-l.html>

- Let  $\theta^*$  be the true parameter, for any  $\varepsilon > 0$ :

$$P(|\hat{\theta} - \theta^*| \geq \varepsilon) \leq 2e^{-2N\varepsilon^2} \leq \delta$$

$$N \geq \frac{1}{2\varepsilon^2} [\ln 2 - \ln \delta]$$

$$N \geq 270; (\varepsilon = 0.1, \delta = 0.01)$$



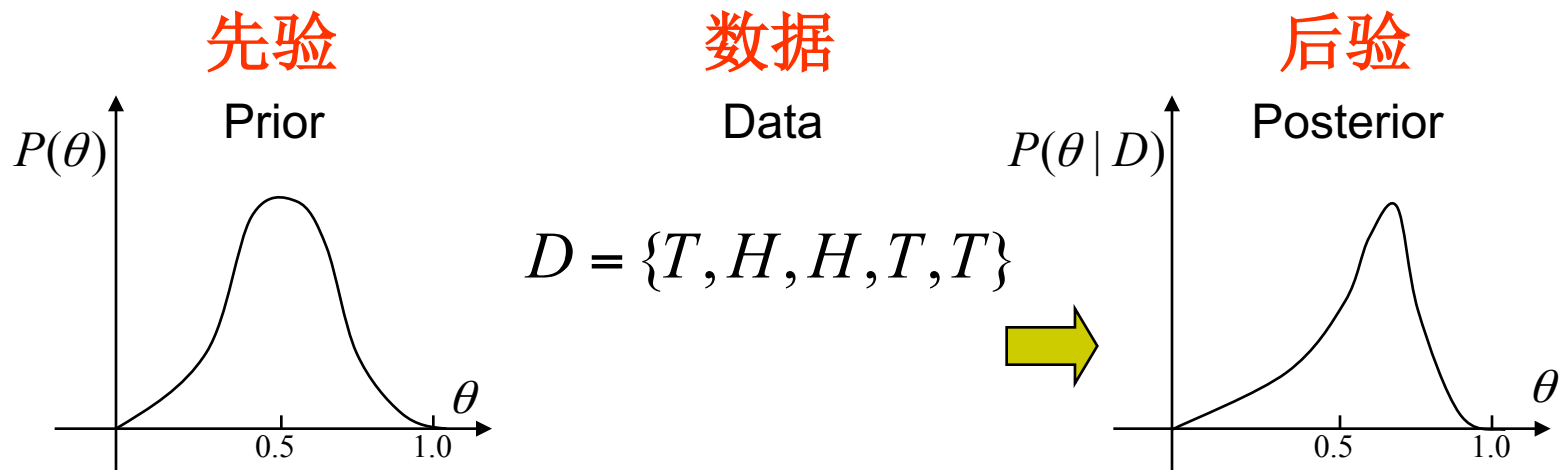
# PAC Learning

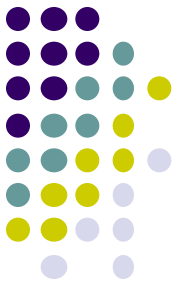
- **PAC: Probably Approximate Correct**
- B: I want to know the thumbtack parameter  $\theta$ , within  $\varepsilon = 0.1$ , with probability at least  $1 - \delta = 0.99$ . How many flips?
- Y: 270, 😊

# Prior: knowledge before experiments



- B: Wait, I know that the thumbtack is “close” to 50-50. What can you ...?
- Y: I can learn it the Bayesian way...
- Rather than estimating a single  $\theta$ , we obtain a distribution over possible values of  $\theta$





# Bayesian Learning

- Bayes rule:

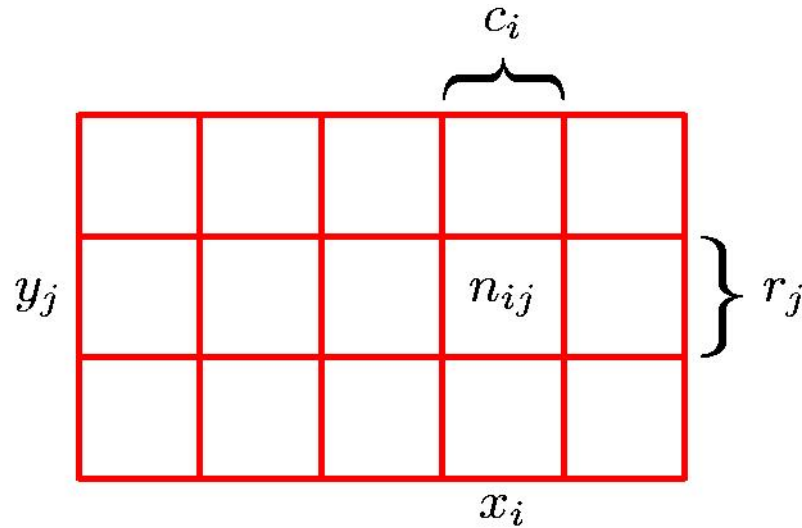
$$\text{Posterior} \rightarrow P(\theta | D) = \frac{\overset{\text{Prior}}{\downarrow} P(\theta) \overset{\text{Likelihood}}{\downarrow} P(D | \theta)}{P(D) \leftarrow \begin{array}{l} \text{Data distribution} \\ \text{evidence} \\ \text{(Normalization)} \end{array}}$$

- Or equivalently:

$$P(\theta | D) \propto P(\theta)P(D | \theta)$$



# Probability Theory



## • Sum Rule

$$\begin{aligned} p(X = x_i) &= \frac{c_i}{N} = \frac{1}{N} \sum_{j=1}^L n_{ij} \\ &= \sum_{j=1}^L p(X = x_i, Y = y_j) \end{aligned}$$

## Product Rule

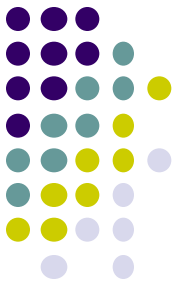
$$\begin{aligned} p(X = x_i, Y = y_j) &= \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N} \\ &= p(Y = y_j | X = x_i) p(X = x_i) \end{aligned}$$



# Probability concepts

- Random variables:  $x$
- Probability (function):  $P(X \leq x)$ ,  $P(x)$
- Density (function):  $f(x)$ ,
- Independency:  $P(x, y) = P(x)P(y)$
- Feature quantities:
  - Mean, expectation  $E(x) = \int x f(x) dx$
  - Covariance
    - $\text{cov}(x, y) = 0$ , uncorrelatedness / irrelevant (统计无关)
  - Higher order moments

# The Rules of Probability



- Sum Rule

$$p(X) = \sum_Y p(X, Y)$$

- Product Rule

$$p(X, Y) = p(Y|X)p(X)$$

# Bayes' Theorem



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

$$p(X) = \sum_Y p(X|Y)p(Y)$$

posterior  likelihood  $\times$  prior





# Bayesian Learning in our case

- Likelihood function is simply Binomial:

$$P(D | \theta) = \theta^{\alpha_H} (1 - \theta)^{\alpha_T}$$

- What about prior?
  - Represent expert knowledge
  - Simple posterior form
- Conjugate priors: (共轭先验)
  - Closed-form representation of posterior
  - For Binomial, conjugate prior is Beta distribution



# Beta prior distribution – $P(\theta)$

Gamma function

- Prior: Beta distribution

$$\Gamma(x+1) = x\Gamma(x), \Gamma(1) = 1$$

$$P(\theta) = \frac{\theta^{\beta_H-1} (1-\theta)^{\beta_T-1}}{B(\beta_H, \beta_T)} \sim \text{Beta}(\theta | \beta_H, \beta_T) = \frac{\Gamma(\beta)}{\Gamma(\beta_H)\Gamma(\beta_T)} \theta^{\beta_H-1} (1-\theta)^{\beta_T-1}$$

- Likelihood: Binomial distribution

$$P(D | \theta) = \theta^{\alpha_H} (1-\theta)^{\alpha_T}$$

- Posterior:

$$\begin{aligned} P(\theta | D) &\propto P(\theta)P(D | \theta) \\ &\propto \theta^{\alpha_H} (1-\theta)^{\alpha_T} \theta^{\beta_H-1} (1-\theta)^{\beta_T-1} \\ &\sim \text{Beta}(\alpha_H + \beta_H, \alpha_T + \beta_T) \end{aligned}$$



# Using Bayesian posterior

- Posterior distribution:

$$P(\theta | D) \sim \text{Beta}(\alpha_H + \beta_H, \alpha_T + \beta_T)$$

- Bayesian inference:

- No longer single parameter:

$$E[f(\theta)] \sim \int_0^1 f(\theta) P(\theta | D) d\theta$$

- Integral, ☹



# Expectation (数学期望)

- Random variable:  $\theta$
- Random function:  $f(\theta)$
- Expectation:

$$E[f(\theta)] \sim \int_0^1 f(\theta) P(\theta | D) d\theta$$

# MAP:

## Maximum a posteriori approximation

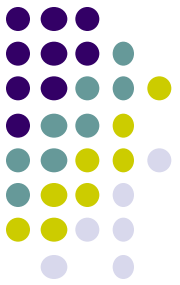


$$P(\theta | D) \sim \text{Beta}(\alpha_H + \beta_H, \alpha_T + \beta_T)$$

$$E[f(\theta)] = \int_0^1 f(\theta) P(\theta | D) d\theta \leftarrow \text{approximation}$$

- MAP: use most likely parameter

$$\hat{\theta} = \arg \max_{\theta} P(\theta | D) \quad E[f(\theta)] \approx f(\hat{\theta}) \leftarrow$$



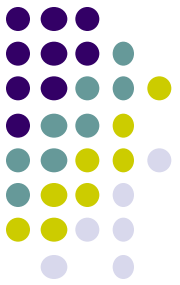
# MAP for Beta distribution

$$P(\theta | D) \sim \text{Beta}(\alpha_H + \beta_H, \alpha_T + \beta_T)$$

- MAP: use most likely parameter

$$\hat{\theta} = \arg \max_{\theta} P(\theta | D) = \frac{\alpha_T + \beta_T - 1}{\alpha_H + \beta_H + \alpha_T + \beta_T - 2}$$

- Beta prior equivalent to extra thumbtack flips
- As  $N = \alpha_T + \alpha_H \rightarrow \infty$ , prior is “forgotten”
- But, for **small sample size**, prior is important!



# More ...

- B: Can we handle more complex cases?
- Y: Yes, :-D
- Prior: a mixture of beta distribution
  - $P(\theta) \sim 0.4\text{Beta}(20,1) + 0.4\text{Beta}(1,20) + 0.2\text{Beta}(2,2)$

# Multinomial distribution



- B: Now if I give you a dice (骰子), then ...
- Y: I can solve this problem in a similar way.
- Likelihood:

$$P(X = x^k \mid \boldsymbol{\theta}) = \theta_k, \quad k = 1, 2, \dots, r,$$

$$\boldsymbol{\theta} = \{\theta_1, \dots, \theta_r\}, \quad \theta_1 + \dots + \theta_r = 1$$

$$D = \{X_1 = x_1, \dots, X_N = x_N\} \Rightarrow \{N_1, \dots, N_r\}$$

$$P(D \mid \boldsymbol{\theta}) = \prod_{i=1}^r \theta_i^{N_i}$$





# Multinomial distribution

- Conjugate prior (Dirichlet distribution):

$$P(\boldsymbol{\theta}) = \text{Dir}(\boldsymbol{\theta} \mid \alpha_1, \dots, \alpha_r) = \frac{\Gamma(\alpha)}{\prod_{k=1}^r \Gamma(\alpha_k)} \prod_{k=1}^r \theta_k^{\alpha_k - 1}, \quad \alpha = \sum_{k=1}^r \alpha_k$$

- Solution:

$$P(X_{N+1} = x^k \mid D) = \int \theta_k \text{Dir}(\boldsymbol{\theta} \mid \alpha_1 + N_1, \dots, \alpha_r + N_r) d\boldsymbol{\theta} = \frac{\alpha_k + N_k}{\alpha + N}$$

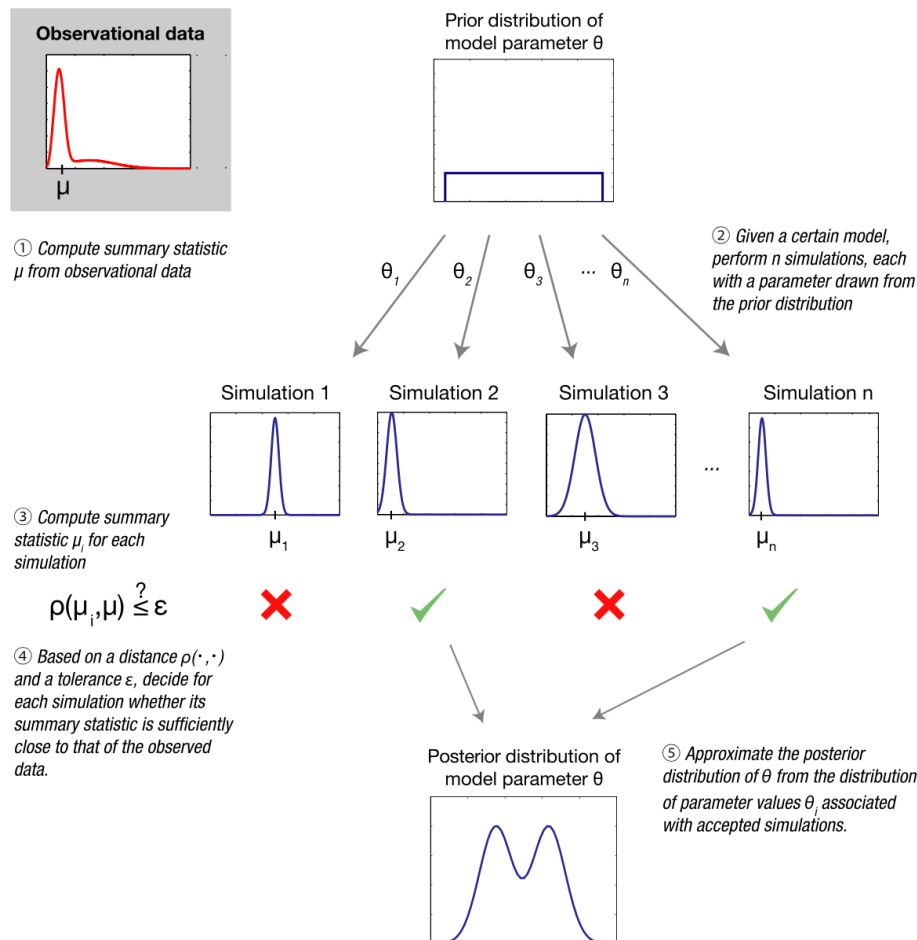
- Important fact:

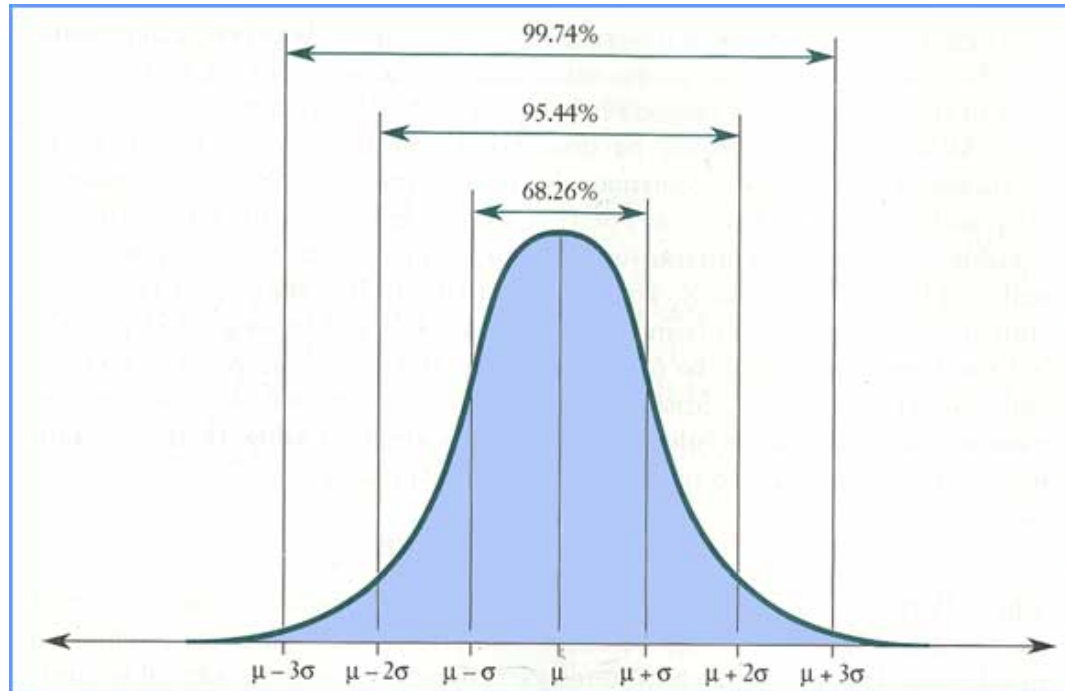
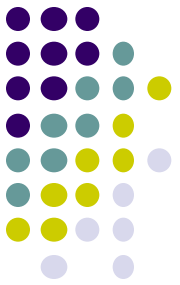
$$P(D) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{k=1}^r \frac{\Gamma(\alpha_k + N_k)}{\Gamma(\alpha_k)}$$

# Beyond MAP: Approximate Bayesian Computation



- Approximate Bayesian Computational methods [[ARXIV](#)]
- Approximate Bayesian Computation (ABC) in practice [[PDF](#)]

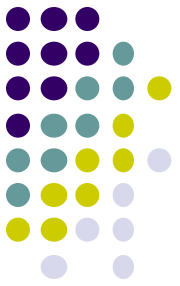




在数据处理中，高斯分布无处不在

# Gaussian distribution

# Gaussian distribution



均值

mean

Continuous random variable:

$$P(x | \mu, \delta) \sim \frac{1}{\underbrace{\delta \sqrt{2\pi}}_{\text{variance}}} e^{-\frac{(x - \mu)^2}{2\delta^2}}$$

variance      Normalization

方差

Consider the difference between continuous and discrete variables?



# MLE for Gaussian

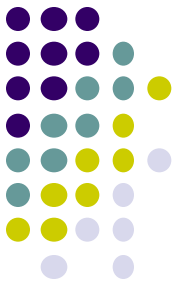
- Prob. of i.i.d. samples  $D = \{x_1, x_2, \dots, x_N\}$

likelihood 
$$P(D | \mu, \sigma) = \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{\frac{-(x_i - \mu)^2}{2\sigma^2}}$$

- The magic of log (to log-likelihood)

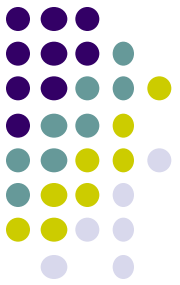
$$\begin{aligned} \ln P(D | \mu, \sigma) &= \ln \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{\frac{-(x_i - \mu)^2}{2\sigma^2}} \\ &= -N \ln(\sigma \sqrt{2\pi}) - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2} \end{aligned}$$

# MLE for mean of a Gaussian



$$\begin{aligned}\frac{\partial}{\partial \mu} \ln P(D | \mu, \sigma) &= \frac{\partial}{\partial \mu} \ln \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{\frac{-(x_i - \mu)^2}{2\sigma^2}} \\ &= \frac{\partial}{\partial \mu} - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2} \\ &= \sum_{i=1}^N \frac{(x_i - \mu)}{\sigma^2} = 0\end{aligned}$$

$$\mu = \frac{1}{N} \sum_i x_i$$



# MLE for variance of a Gaussian

$$\begin{aligned}\frac{\partial}{\partial \sigma} \ln P(D | \mu, \sigma) &= \frac{\partial}{\partial \sigma} \ln \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{\frac{-(x_i - \mu)^2}{2\sigma^2}} \\ &= \frac{\partial}{\partial \sigma} [-N \ln \sigma \sqrt{2\pi}] - \sum_{i=1}^N \frac{\partial}{\partial \sigma} \left[ \frac{(x_i - \mu)^2}{2\sigma^2} \right] \\ &= -\frac{N}{\sigma} + \sum_{i=1}^N \frac{(x_i - \mu)^2}{\sigma^3} = 0\end{aligned}$$

$$\sigma^2 = \frac{1}{N} \sum_i (x_i - \mu)^2$$

# Gaussian parameters learning



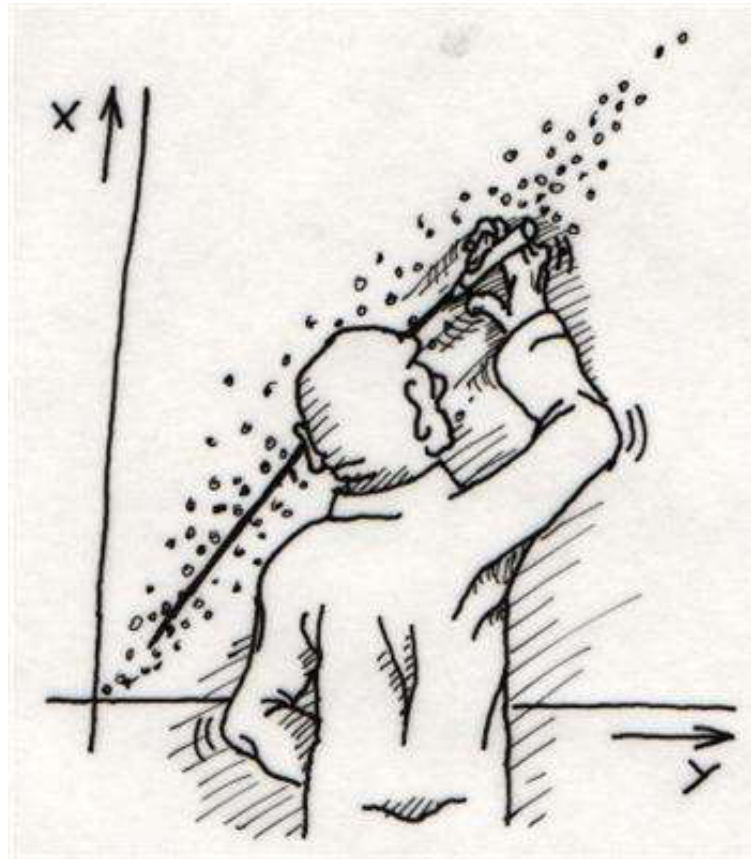
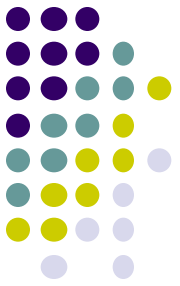
- MLE

$$\hat{\mu} = \frac{1}{N} \sum_i x_i$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_i (x_i - \mu)^2$$

- Bayesian learning: prior?
- Conjugate priors:
  - Mean: Gaussian priors
  - Variance: Wishart Distribution

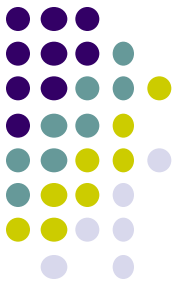




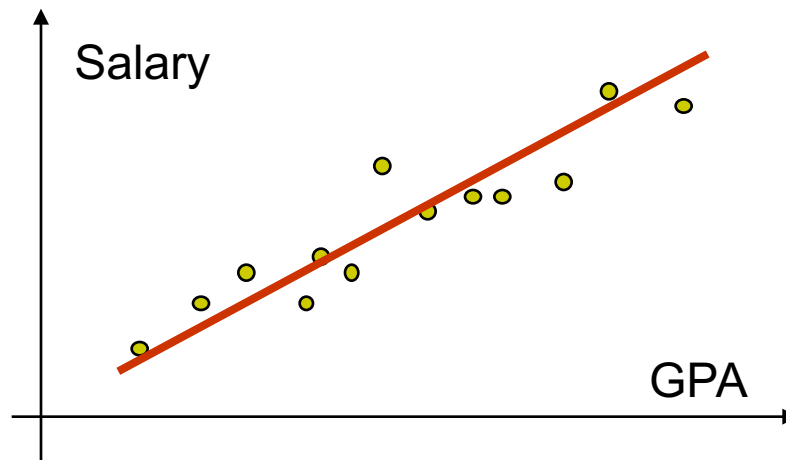
回归的思想，对于理解数据本身的特点与规律，是一种利器

# Regression problems

# Prediction of continuous variable



- B: Wait, that's not what I meant!
- Y: Chill out, dude.
- B: I want to predict a continuous variable for continuous inputs: I want to predict salaries from GPA.
- Y: I can regress that...

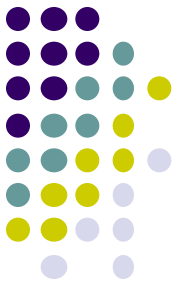




# The regression problem

- **Instances:**  $\langle \mathbf{x}_i, t_i \rangle$
- **Learn:** mapping from  $\mathbf{x}$  to  $t(\mathbf{x})$ .
- **Hypothesis space:**  $t(\mathbf{x}) \approx \hat{f}(x) = \sum_{i=1}^k w_i h_i$ 
  - Given, basis functions  $H = \{h_1, \dots, h_k\}$
  - Find coefficients  $\mathbf{w} = \{w_1, \dots, w_k\}$
- **Problem formulation:**

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_j [t(\mathbf{x}_j) - \sum_{i=1}^k w_i h_i(x)]^2$$



# But, why sum squared error?

- Model:

$$P(t \mid \mathbf{x}, \mathbf{w}, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-[t - \sum_i w_i h_i(x)]^2}{2\sigma^2}}$$

- Learn  $\mathbf{w}$  using MLE

# Maximizing log-likelihood

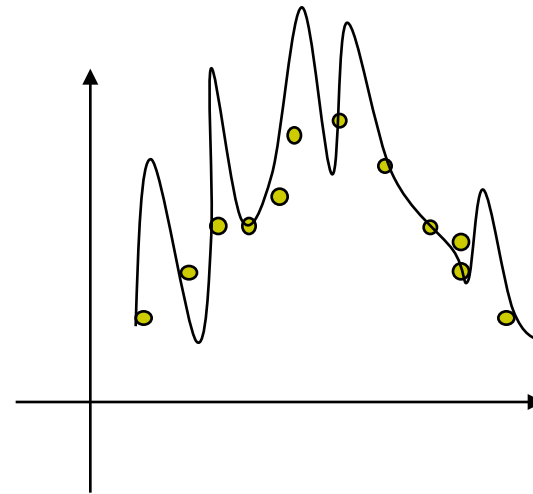
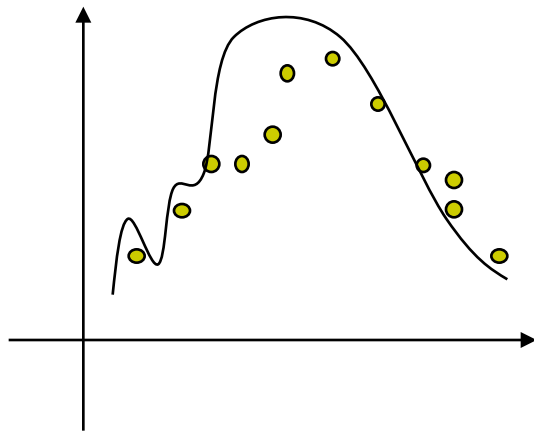


$$\ln P(D \mid \mathbf{w}, \sigma) = \ln \prod_j \left( \frac{1}{\sigma \sqrt{2\pi}} e^{\frac{-[t_j - \sum_i w_i h_i(x_j)]^2}{2\sigma^2}} \right)$$
$$\Rightarrow \min \sum_j \frac{-[t_j - \sum_i w_i h_i(x_j)]^2}{2\sigma^2}$$

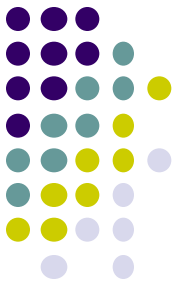


# Bias-Variance Tradeoff

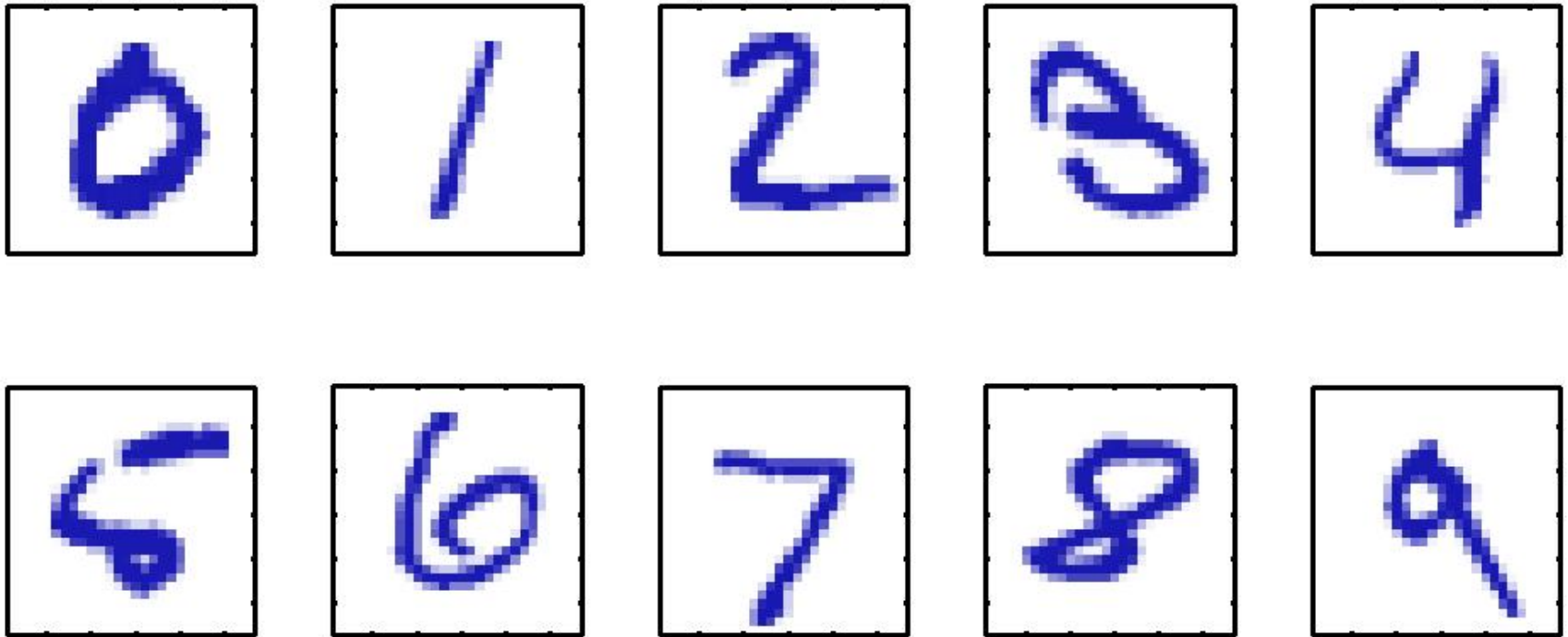
- Choice of hypothesis basis introduce learning bias:
  - More complex basis:
    - Less bias
    - More variance (over-fitting)



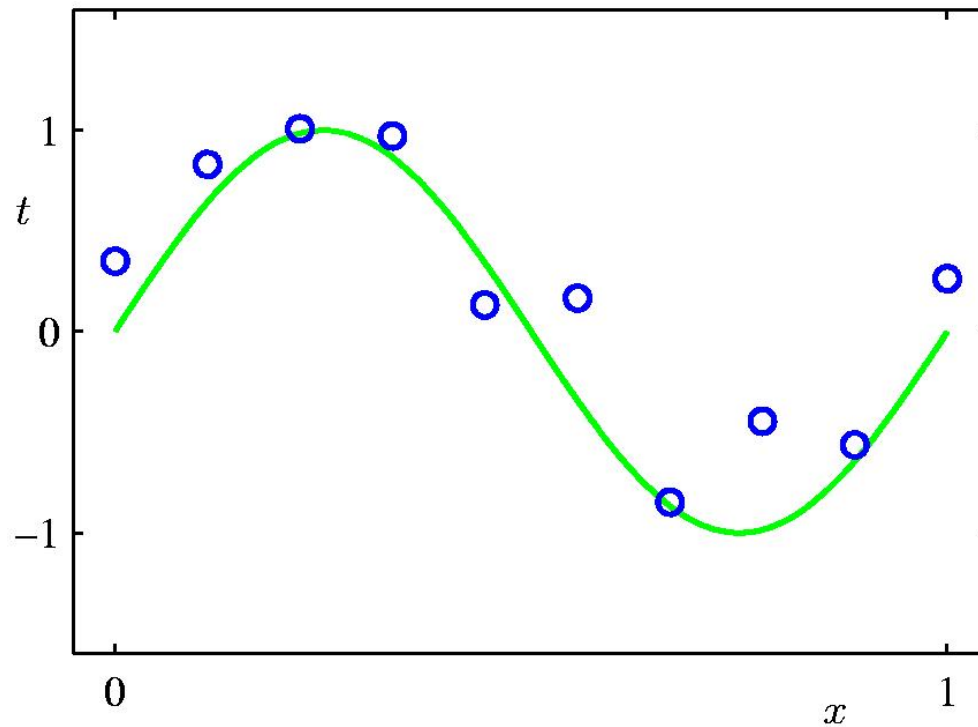
# Example



## Handwritten Digit Recognition



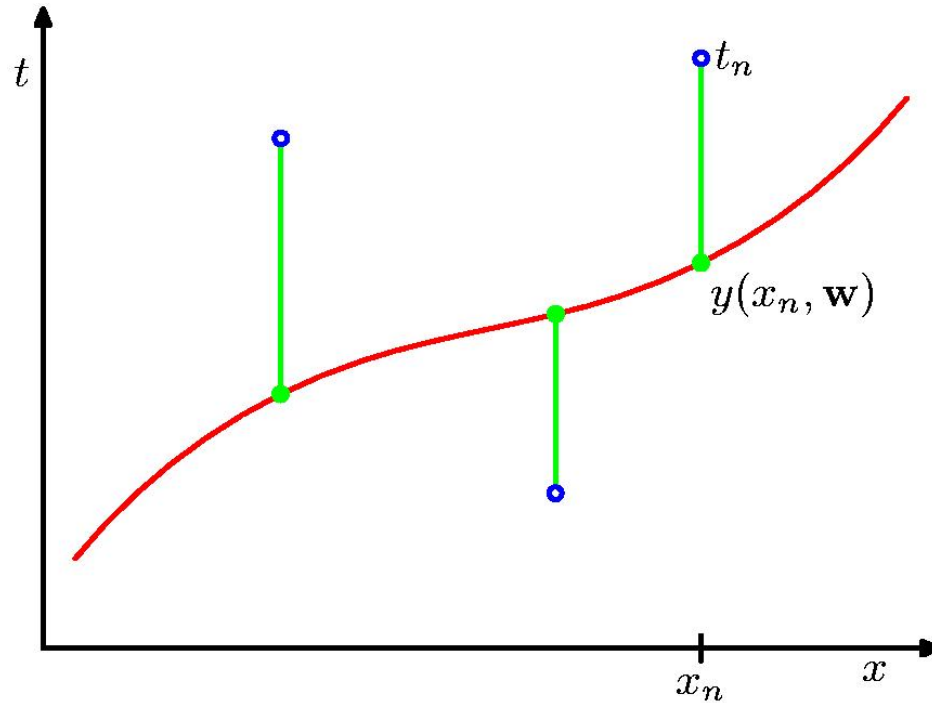
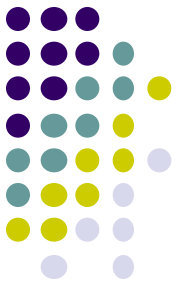
# Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

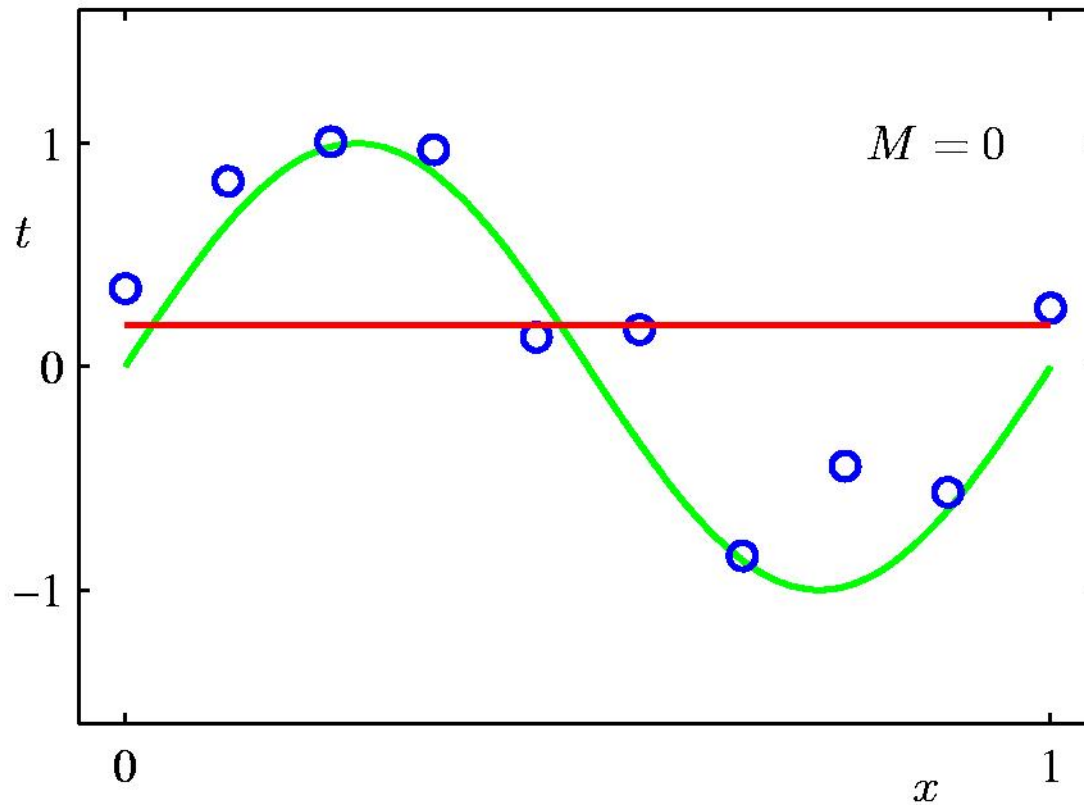
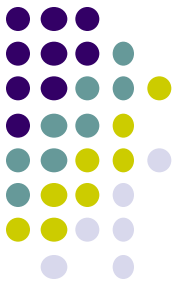


# Sum-of-Squares Error Function

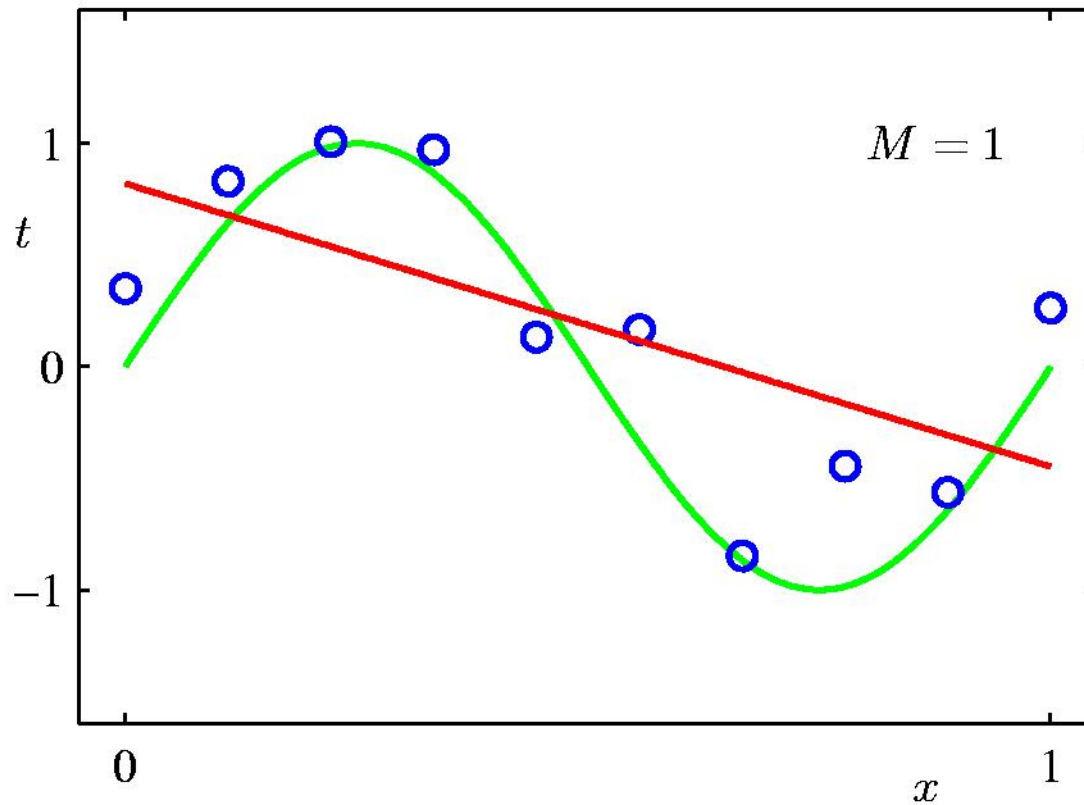


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$$

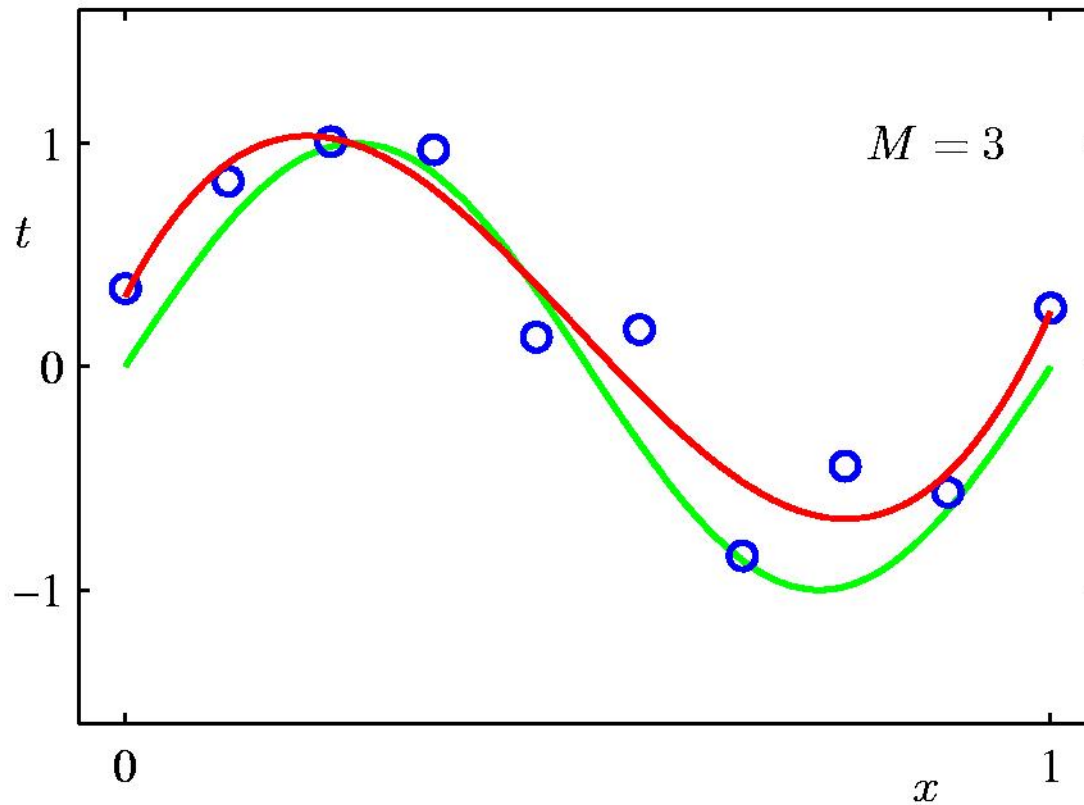
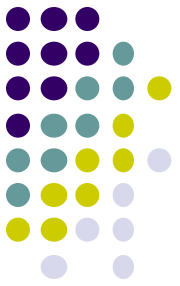
# 0<sup>th</sup> Order Polynomial



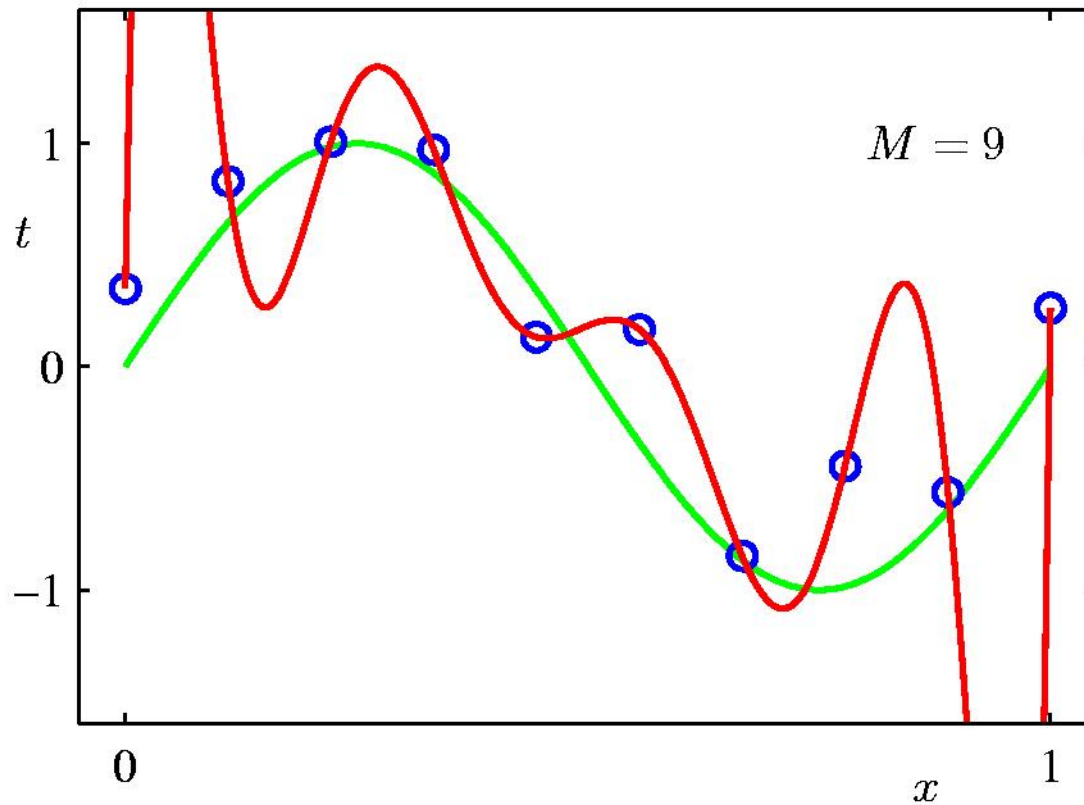
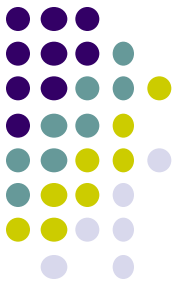
# 1<sup>st</sup> Order Polynomial



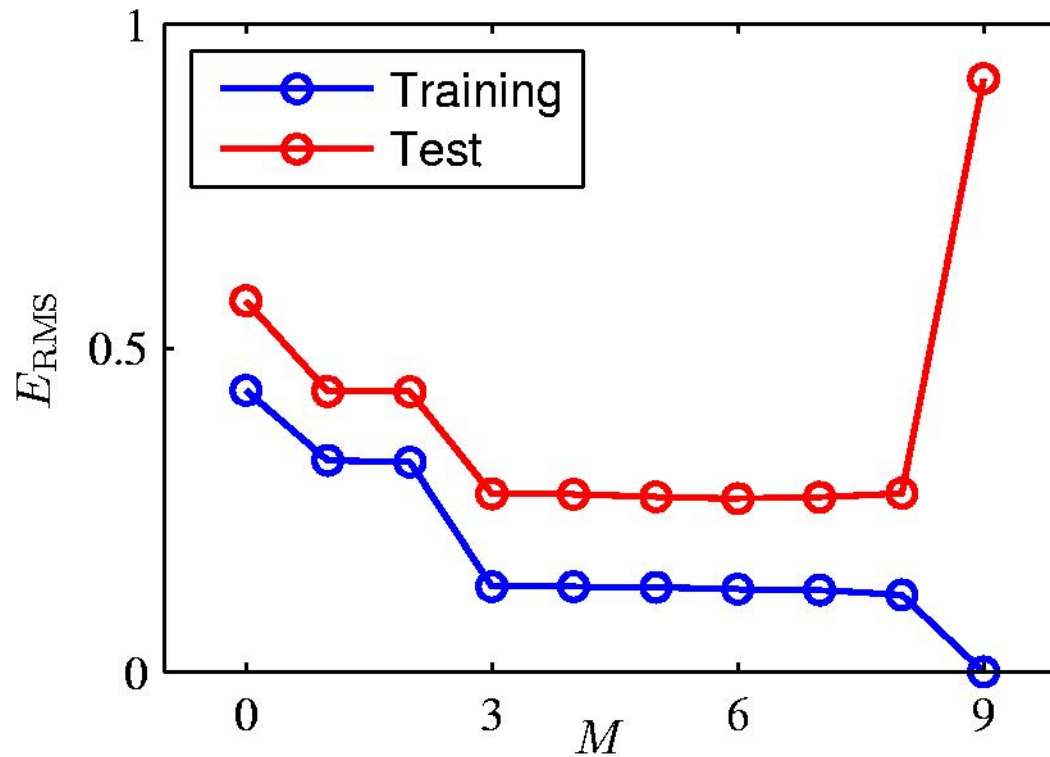
# 3<sup>rd</sup> Order Polynomial



# 9<sup>th</sup> Order Polynomial

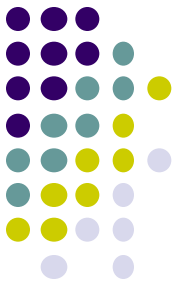


# Over-fitting



Root-Mean-Square (RMS) Error:  $E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$

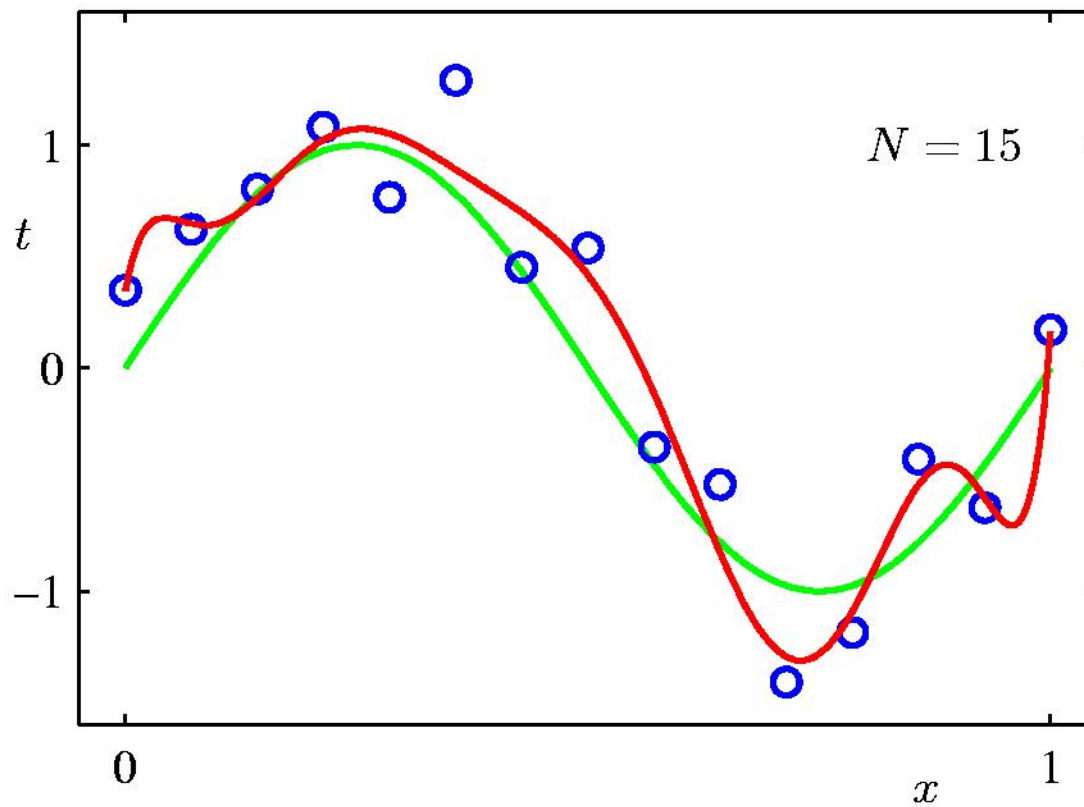
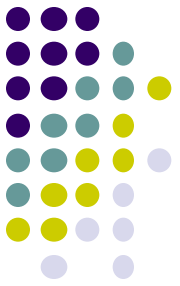
# Polynomial Coefficients



	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^*$	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

# Data Set Size: $N = 15$

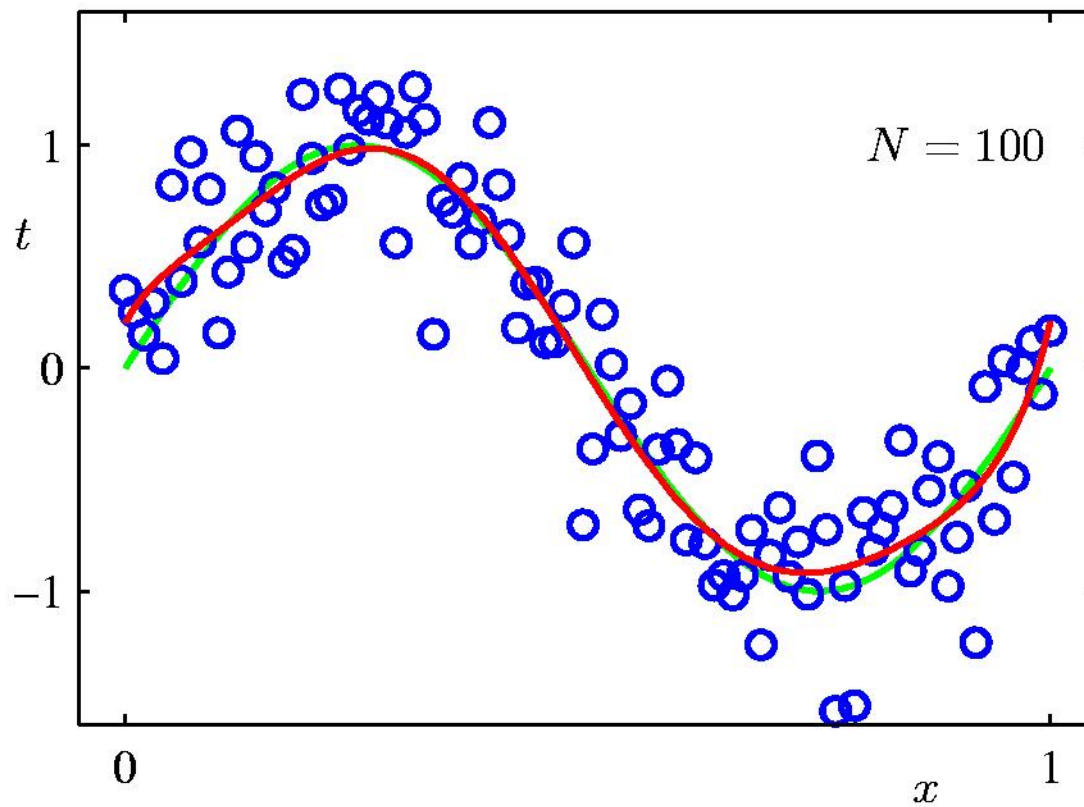
## 9<sup>th</sup> Order Polynomial



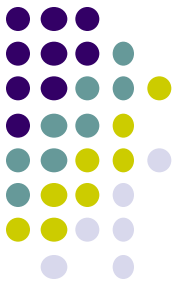


# Data Set Size: $N = 100$

## 9<sup>th</sup> Order Polynomial



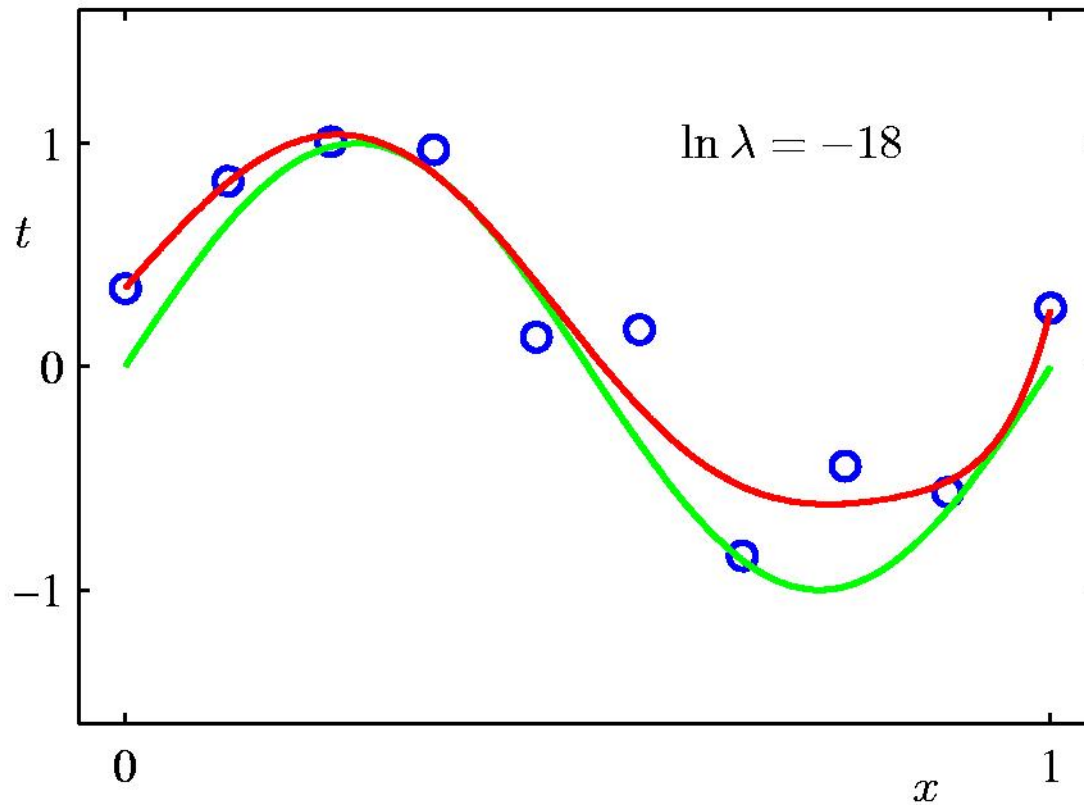
# Regularization



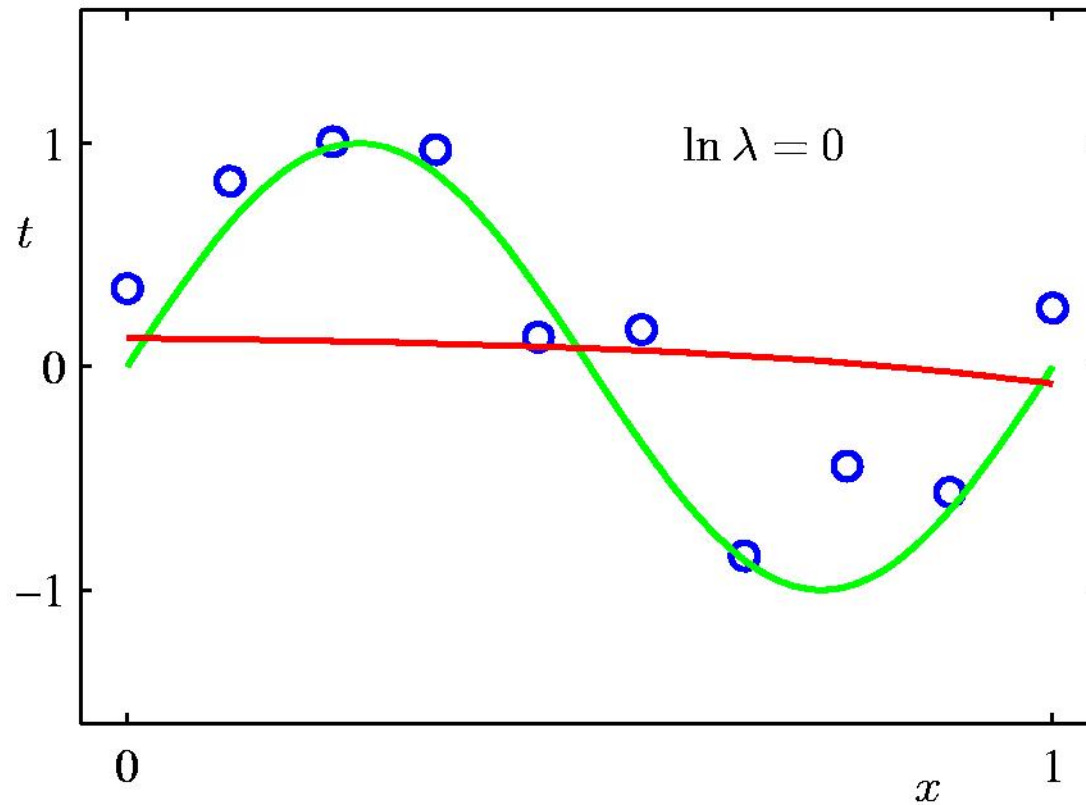
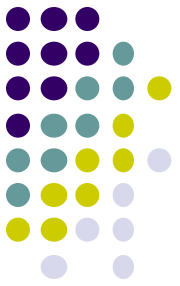
- Penalize large coefficient values

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

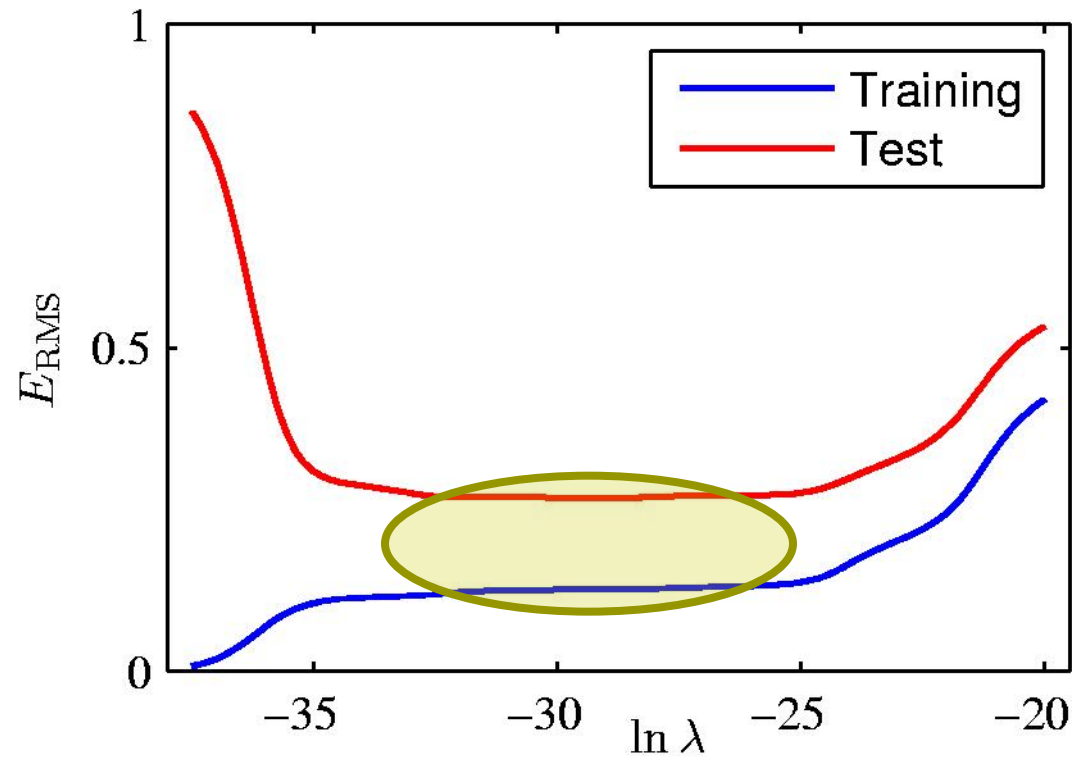
# Regularization: $\ln \lambda = -18$



# Regularization: $\ln \lambda = 0$



# Regularization: $E_{\text{RMS}}$ vs. $\ln \lambda$





# What you need to know

- Point estimation:
  - Maximal Likelihood Estimation
  - Bayesian learning
  - Maximal a Posterior
- Gaussian estimation
- Regression
  - Basis function = features
  - Optimizing sum squared error
  - Relationship between regression and Gaussians
- Bias-Variance trade-off



# Homework01

- Python programming
  - 1-D regression
  - Learn TensorFlow
- Finish the “Gaussian parameters learning”
  - Please use google,  $\wedge\_*$

# The End

新浪微博: @浙大张宏鑫

微信公众号:

