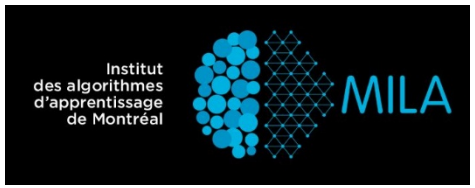
Machine Learning Basics

Jian Tang

tangjianpku@gmail.com



Evaluation

- **Course Projects**:
  - Students should work on course projects in teams (at most 4 students).
  - At the end of this course, each team should make a **presentation** (30%, 20 minutes) and also hand in a project report (70%, due in two weeks after the course is finished).
- **Project report**
  - Should give a clear definition of the problem (10%)
  - A detailed survey of the problem (25%)
  - A proposal (35%)
  - Some preliminary results (not required, + 10 %)
  - Five pages in total (NIPS format, English)

Course Outline

- Introduction & Mathematics (Day 1)
- Machine Learning Basics (Day 2)
- Feedforward Neural Networks & Optimization Tricks (Day 2 & 3)
- Convolutional Neural Networks (Day 3)
- Recurrent Neural Networks (Day 3)
- Deep Learning for Natural Language Understanding (Day 4)
- Graph Representation Learning (Day 5)
- Presentation Session (Day 6)

Schedule Today

- Lecture 1
- Discussion session 1: project discussion
- Lecture 2
- Discussion session 2: project discussion
- Lecture 3

What is Machine Learning?

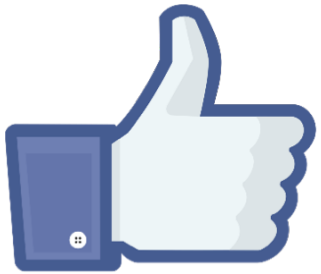
"**Machine learning** is a field of computer science that often uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed."

-Wikipedia

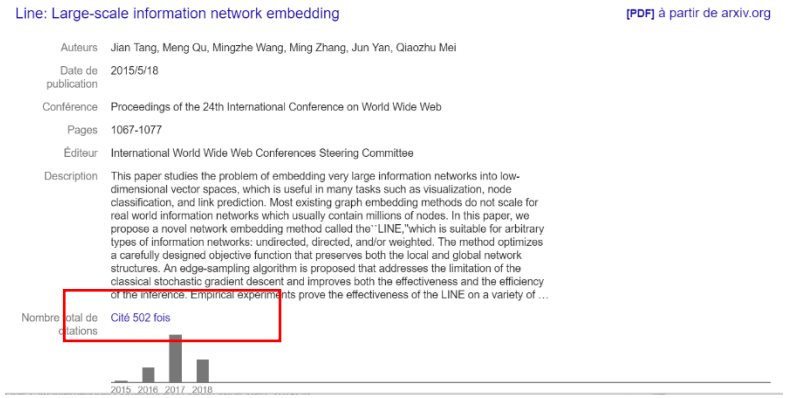
-From https://www.youtube.com/watch?v=2QgyH29x0_M

Regression

- Learning a function f: $R^n \rightarrow R$
- The goal is to accurately predict the target values



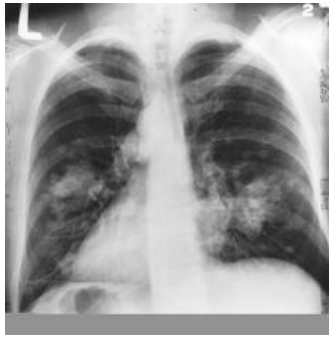
X: message features
Y: the number of likes



X: paper features
Y: the number of citations

Classification

- Learning a function f: $R^n \rightarrow \{1, \ldots, k\}$
- Accurately predict the category of the input data



X: set of pixel intensities
Y: cancer present/cancer absent

Most Helpful Customer Reviews

56 of 63 people found the following review helpful

★★★★☆ **Can A Reference Book Be Too Thorough?**

By B.L. on January 9, 2011

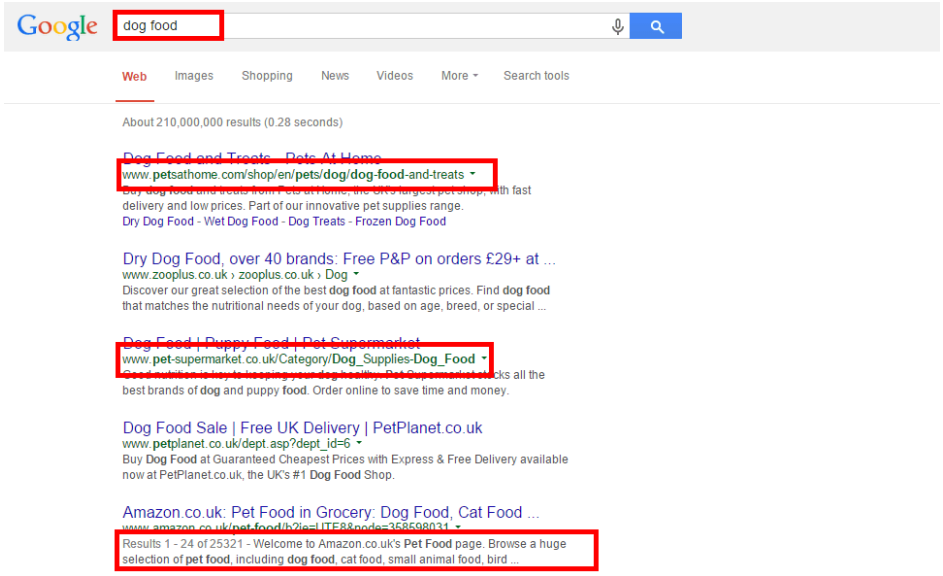
Format: Paperback

Programming Python is a book designed to take people who know Python and guide them on how to actually make it do things in the real world. It's important to note that the material in here (In the December 2010 4th edition) is for 3.X versions of Python and only deals with 2.X to the extent that the versions overlap, so you'll be better off with an earlier edition of the book (or another book designed to deal thoroughly with both versions) if you're working on a project that needs to work uusing earlier versions of Python.

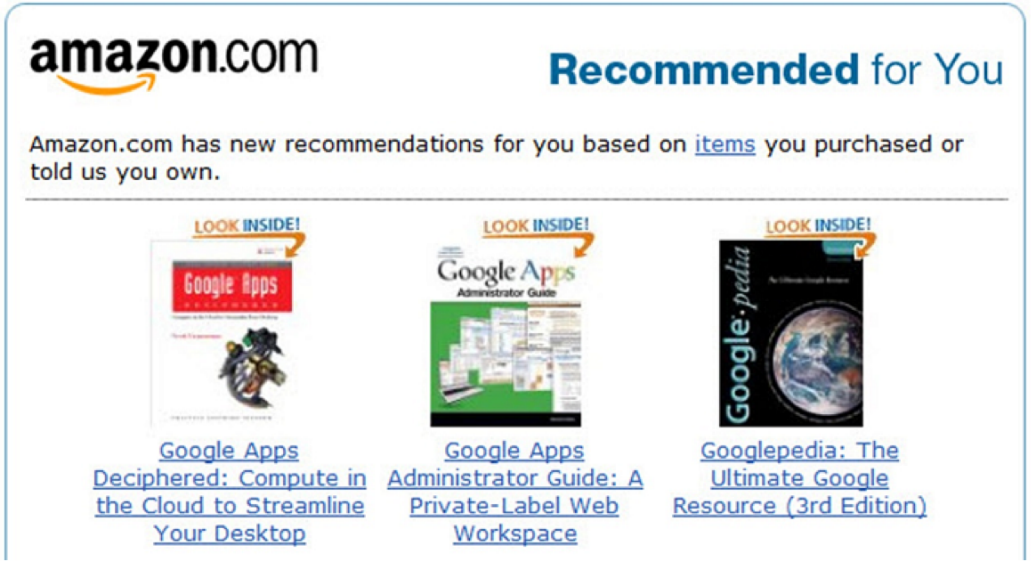
X: user reviews
Y: positive/neural/negative

Ranking

- Learning a function f: $R^n \rightarrow R$
- The goal is to rank the items according to the regression values



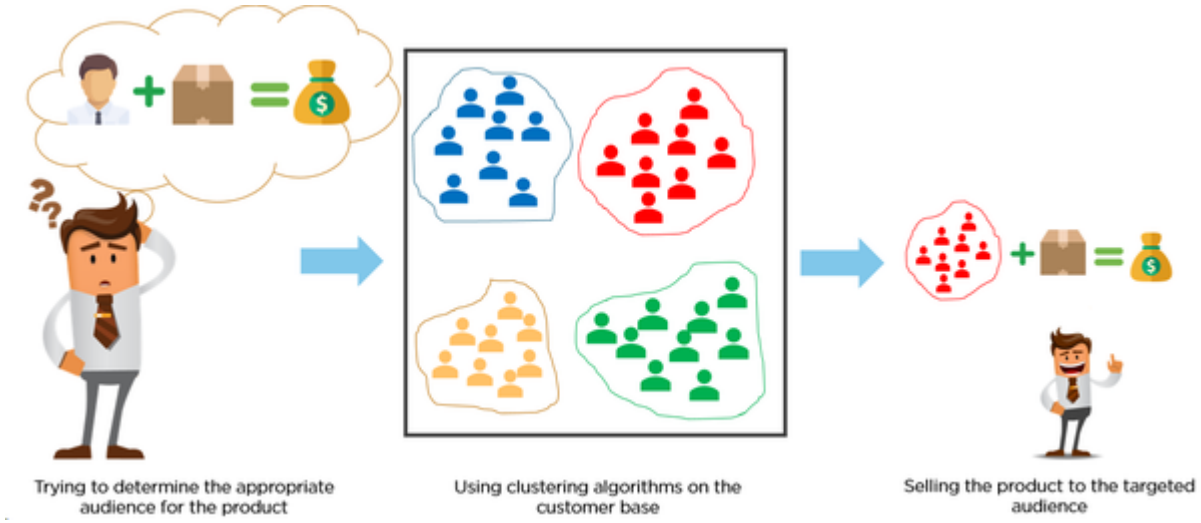
Web Search



Recommender Systems

Clustering

- Group the data into different clusters
- But the clusters are not predefined.



User segmentation

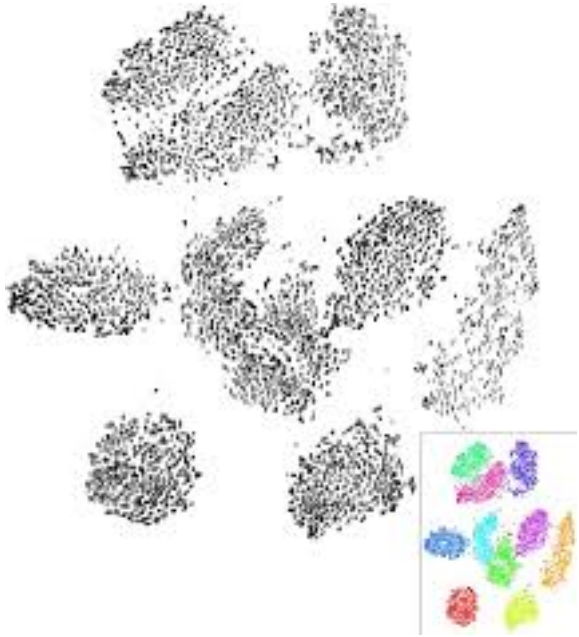
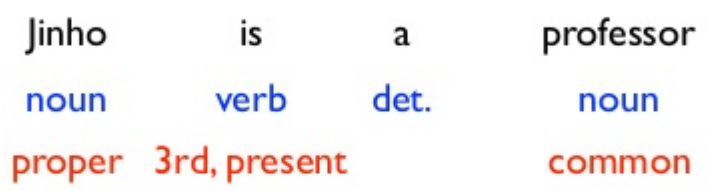


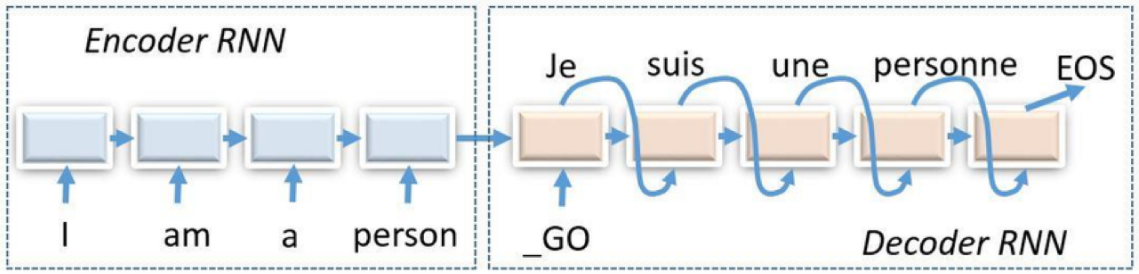
Image clustering

Structured Output

- There are multiple values of the outputs with structured relationships between them



Part-of-speech tagging



Machine translation

Three Different Categories of Machine Learning Algorithms

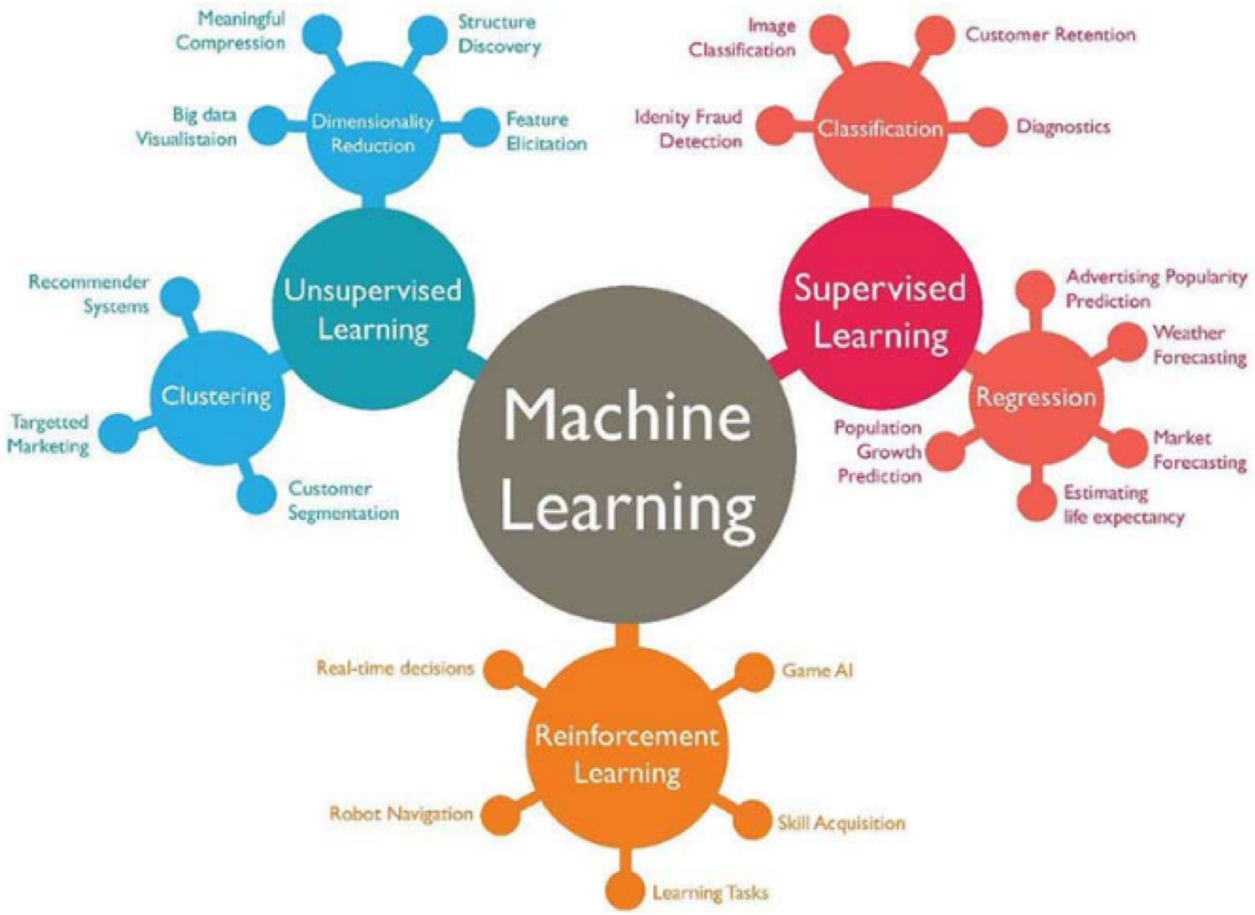


Image from Internet

- Data with labels
  - Classification & regression
- We have training set for which we given right answer for every trainin algorithm
- Training examples contains all the right answers
- Job- to replicate the right answers

Three Different Categories of Machine Learning Algorithms

- Data without labels
  - Clustering
  - Dimension reduction
- Find structures within data

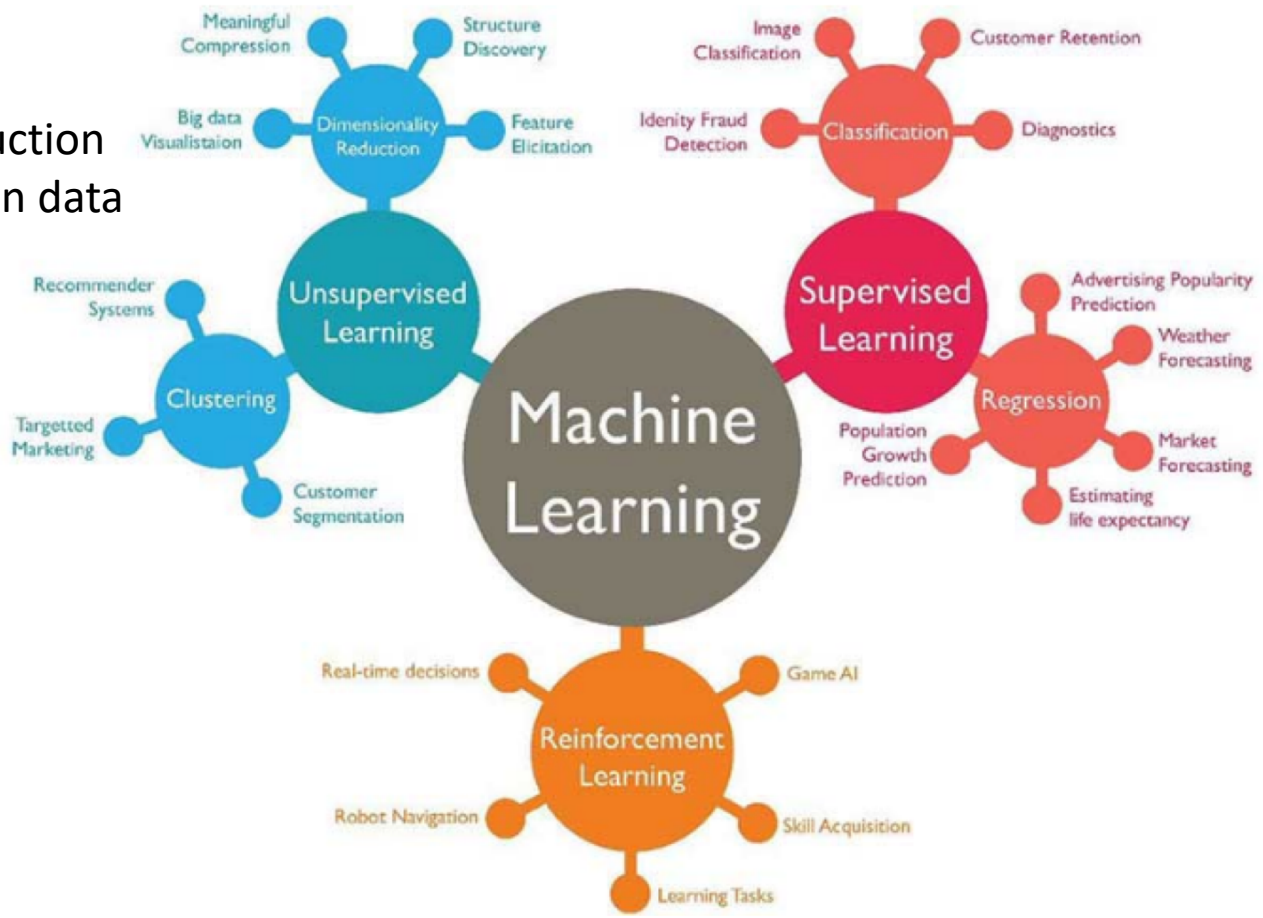


Image from Internet

Three Different Categories of Machine Learning Algorithms

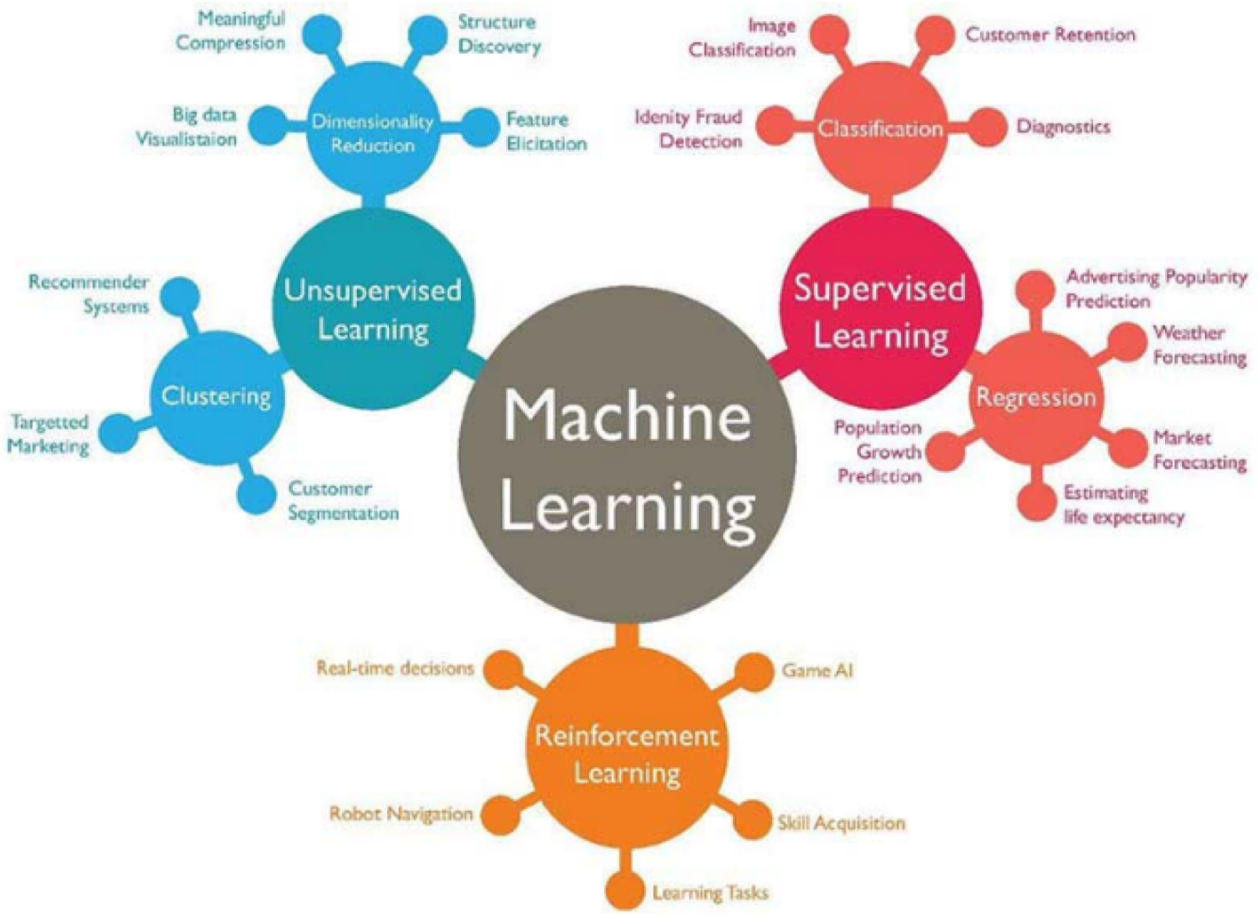


Image from Internet

- Sequential decision process
- Trial-and-error paradigms
- We do not have a target variable
- We have reward signals
- Perform an action and receive a reward from the environment
  - the game Go
- Agents need to plan the path to obtain maximal reward

Outline: Linear Regression

- What is Linear Regression
- Model Capacity, Overfitting, and Undefitting
- Regularization
- Cross Validation
- Decision Theory

Linear Regression

- Given a vector of d-dimensional inputs $\boldsymbol{x} = (x_1, x_2, \dots, x_d)^T$, we want to predict the target (response) using the linear model:

$$y = w_0 + w_1 x_1 + \cdots + w_d x_d = w_0 + \sum_{j=1}^{d} w_j x_j ,$$

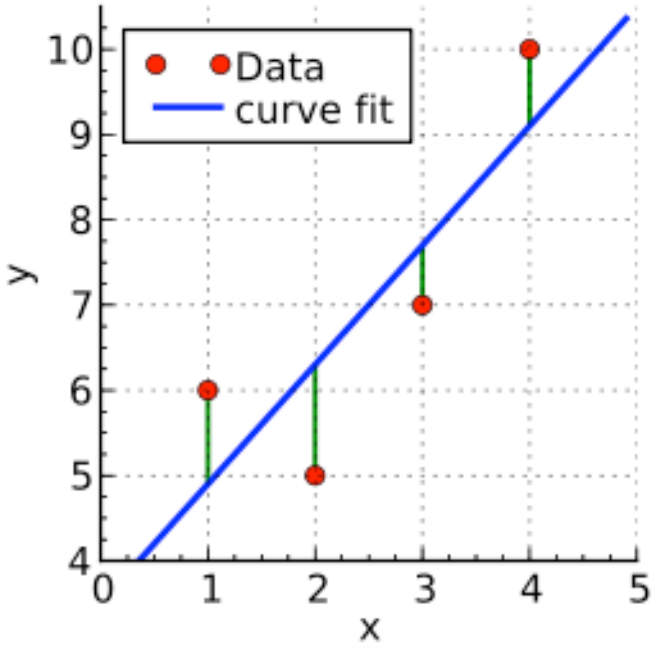
- where $\boldsymbol{w} \in \boldsymbol{R^{d+1}}$ is a vector of **parameters**, w_0 is the intercept.

$$y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{x^T w}$$

- Training set: a set of training data points $X = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N})^T$ and their labels $t = (t_1, t_2, \dots, t_N)^T$
  - the goal is to learn the parameter $\boldsymbol{w}$

Least Square Regression (LSR)

- Training criteria: minimizing the sum of the errors between the predictions by the model $y(\boldsymbol{x}_n, \boldsymbol{w})$ and the targets t_n:

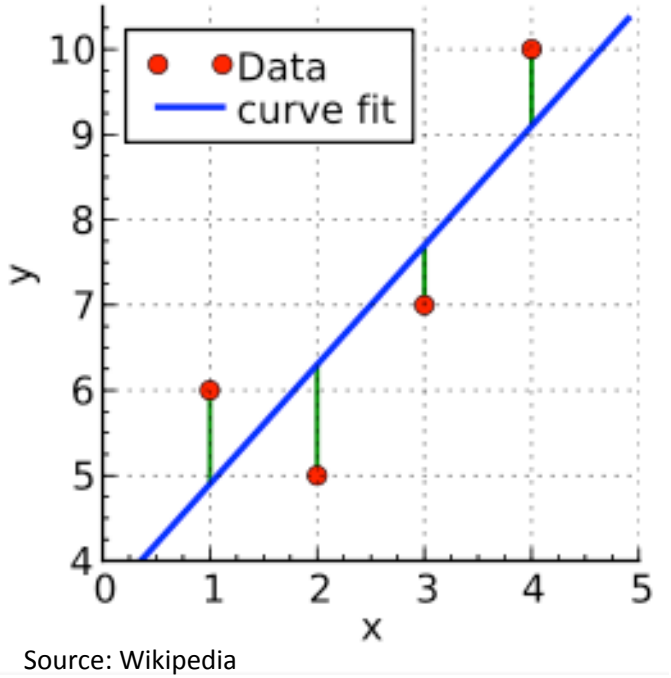


Source: Wikipedia

$$E(w) = 1/2 \sum_{n=1}^{N} (y(\boldsymbol{x}_n, \boldsymbol{w}) - t_n)^2$$

Least Square Regression (LSR)

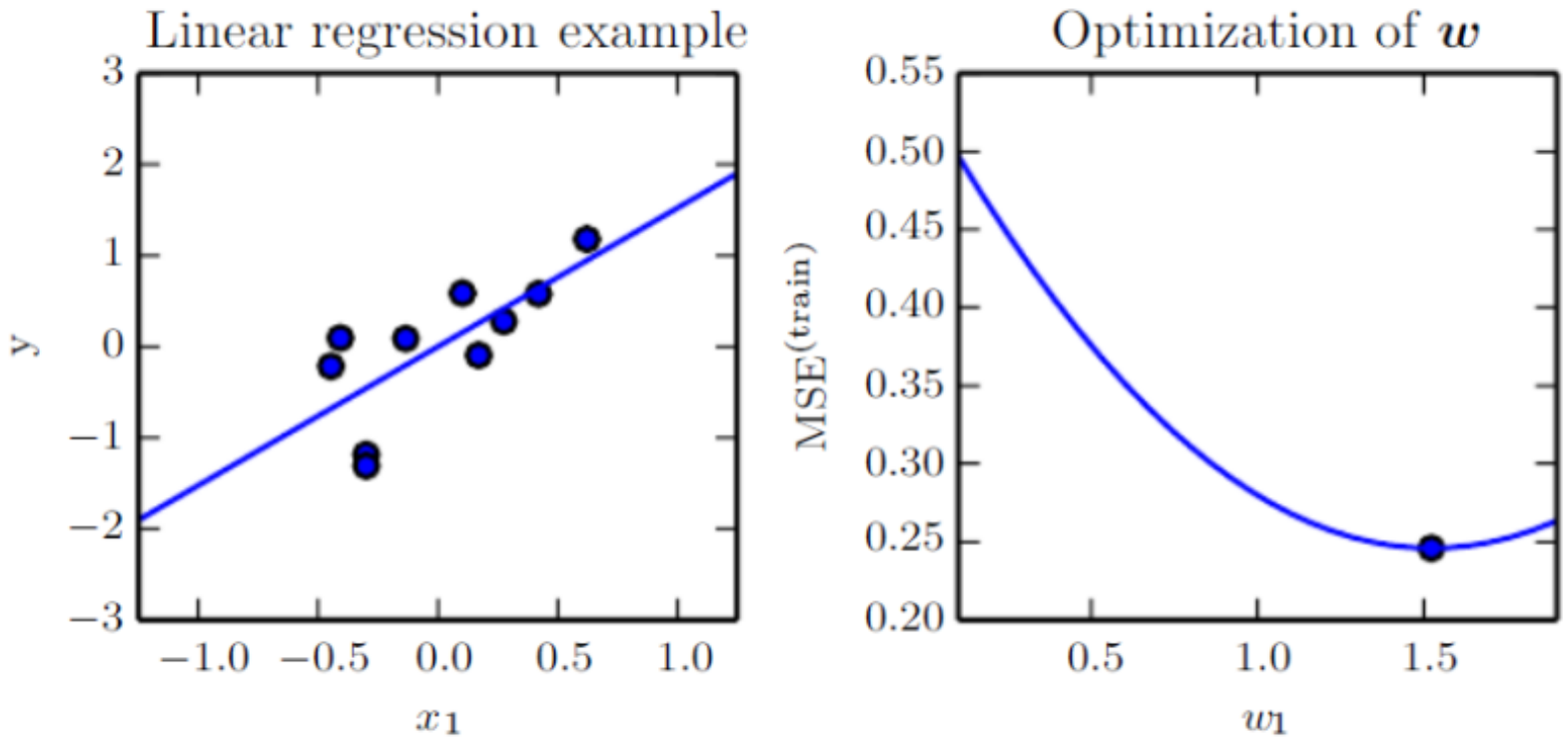
- Unique solutions if $X^T X$ is nonsingular



Source: Wikipedia

$$w^* = (X^T X)^{-1} X^T t$$

Example (with only one input feature)



Outline: Linear Regression

- What is Linear Regression
- Model Capacity, Overfitting, and Undefitting
- Regularization
- Cross Validation
- Decision Theory

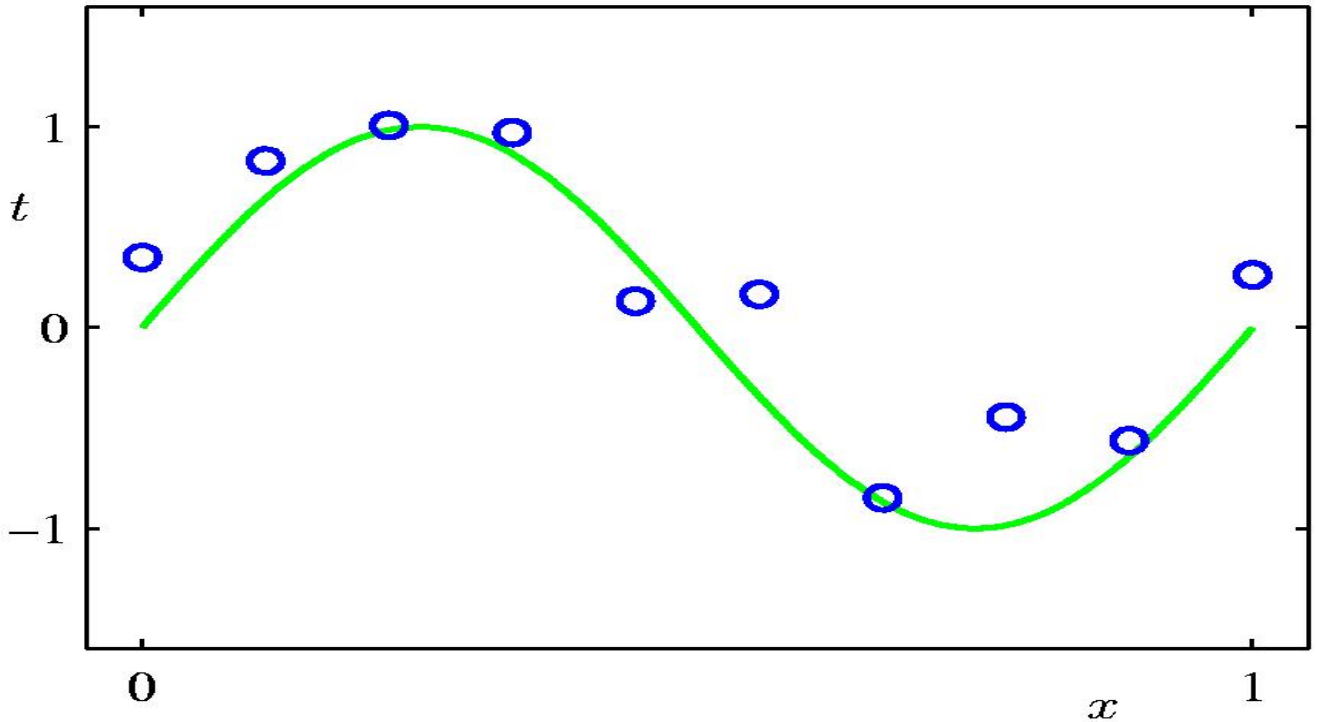
Model Capacity, Underfitting, and Overfitting

- The goal of machine learning model is to maximize the **generalization** ability
  - Perform well on previously unobserved inputs
- Training data => training error
- Test data => test error (generalization error)
- For linear regression:
  - Train the model by minimizing the train error
  - Evaluate the performance of the model according to the test error

Model Capacity, Underfitting, and Overfitting

- Model capacity: the ability to fit a variety of functions
  - Models with more parameters usually have larger capacity
- Underfitting: model is not able to obtain a sufficiently low error value on the training set
- Overfitting: perform wells on training data but not on the test data

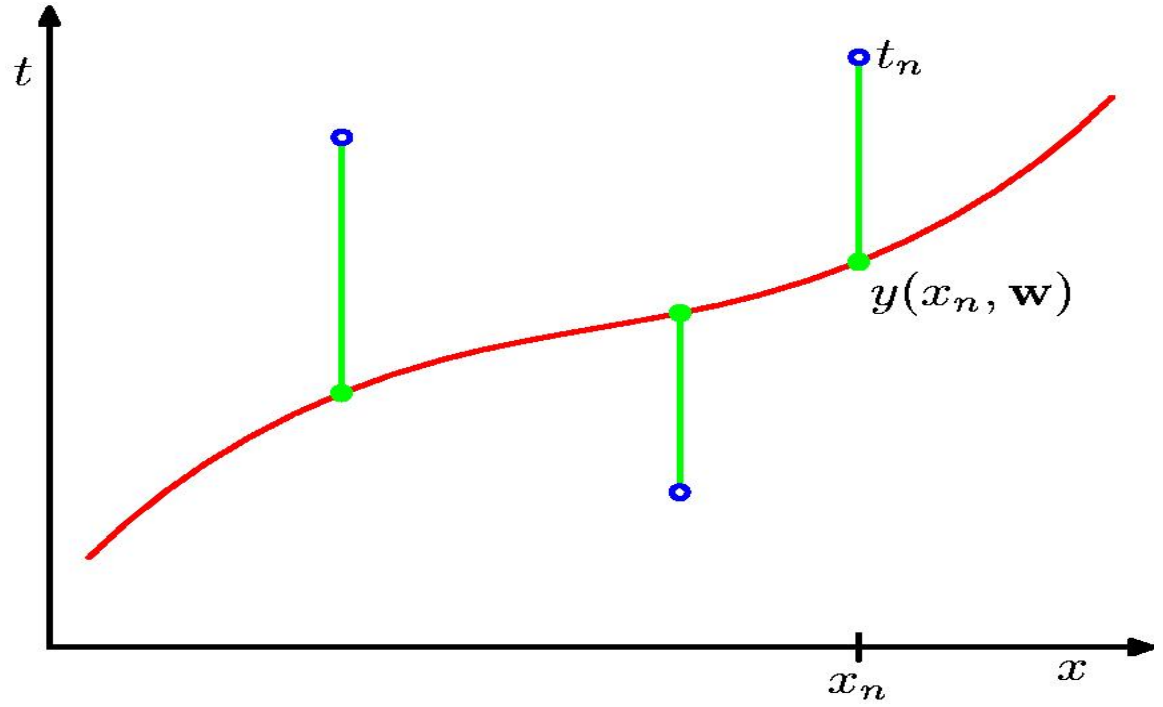
Example: Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

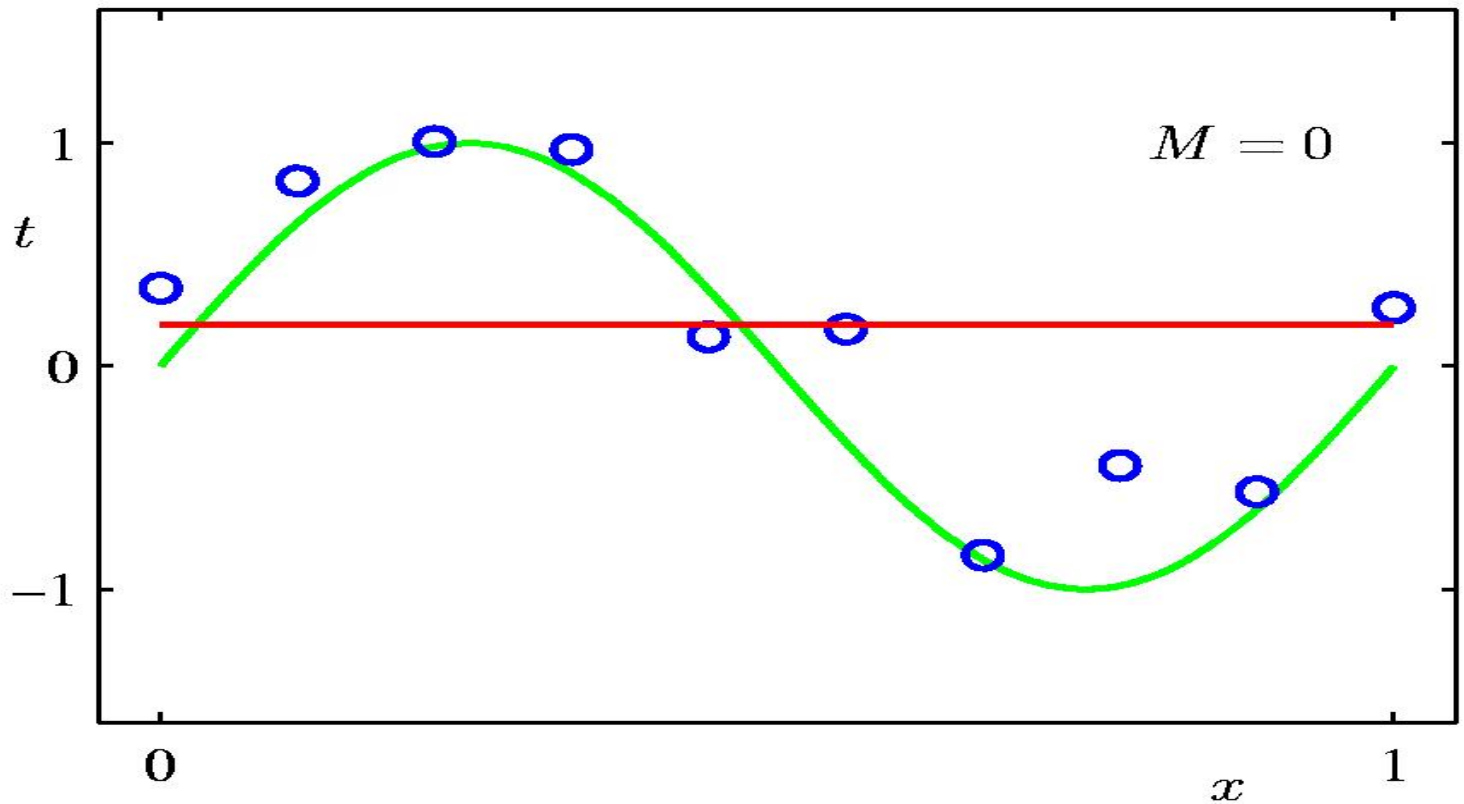
- Given a 1-dimensional observation x, predict a scale t?
- The model is a **linear** function of the coefficients of **w**

Sum-of-Squares Error Function

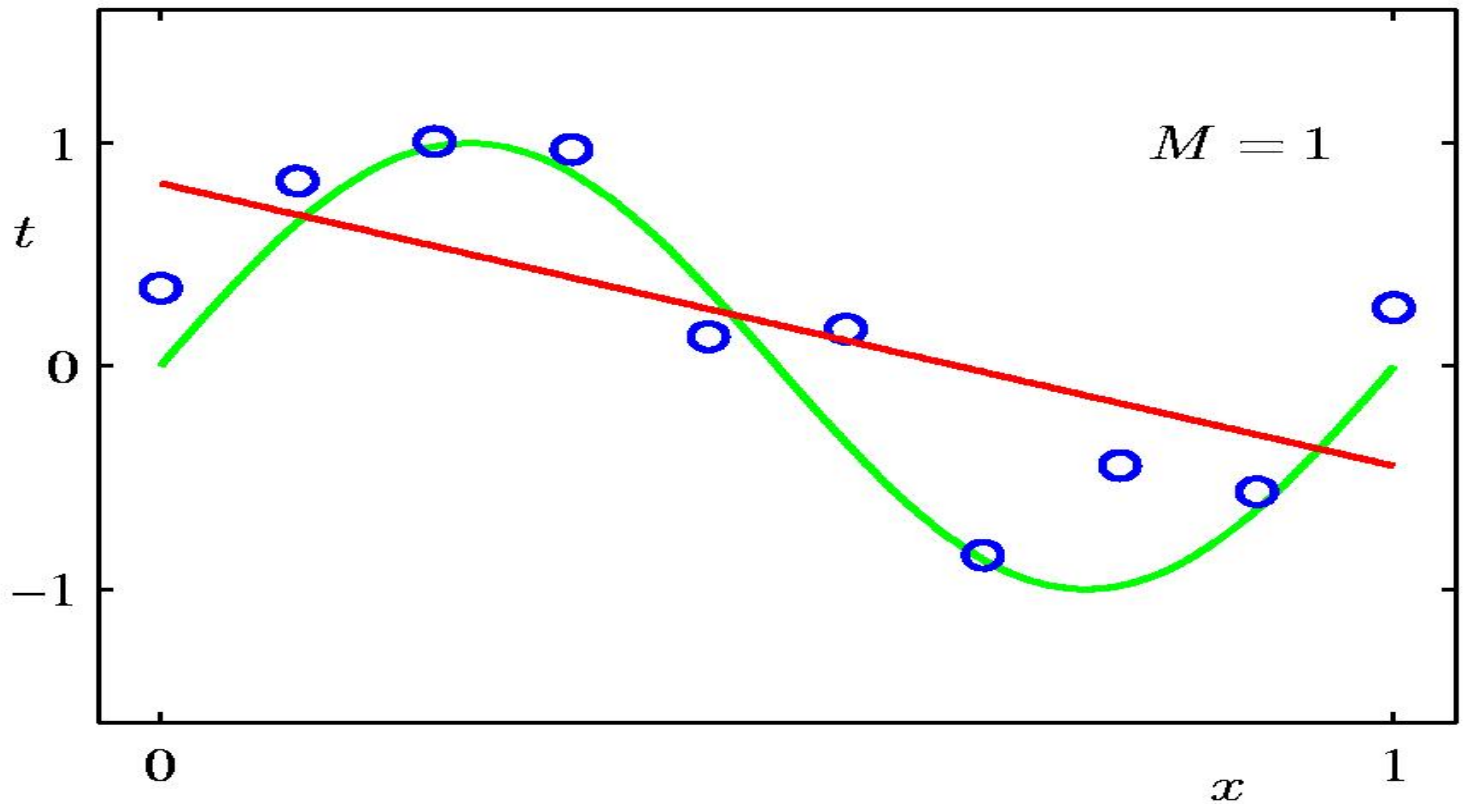


How to choose M?

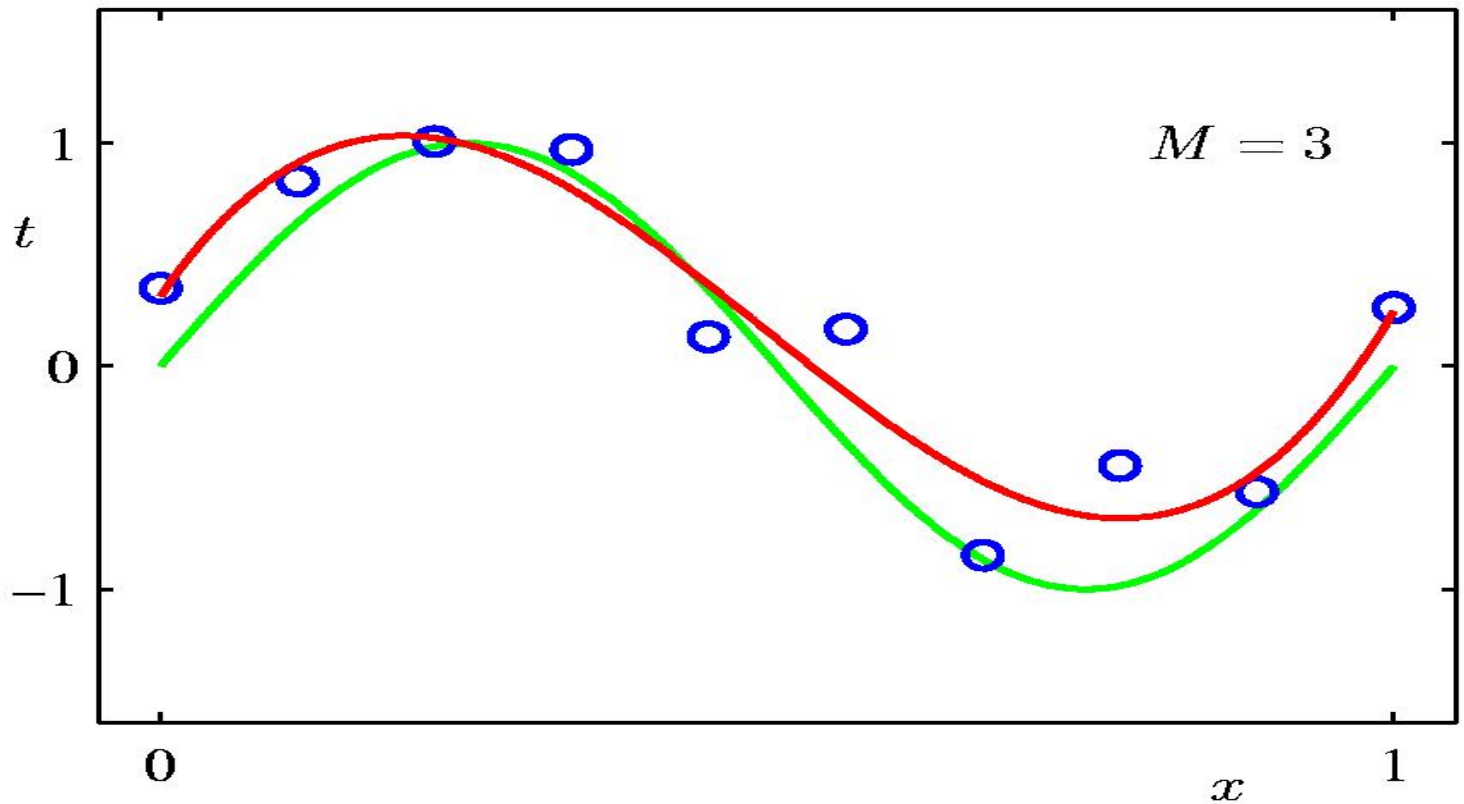
M = 0



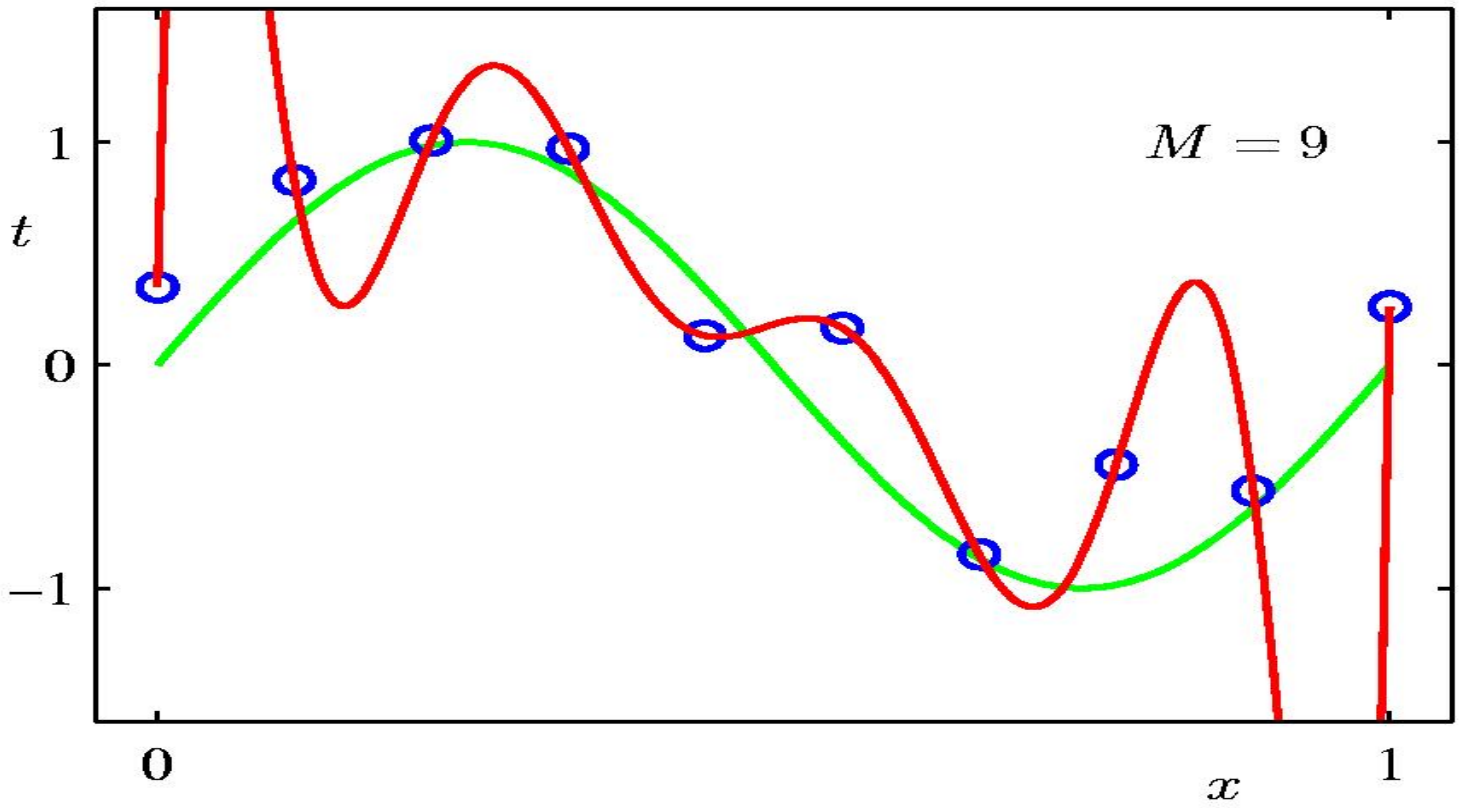
M = 1



M = 3



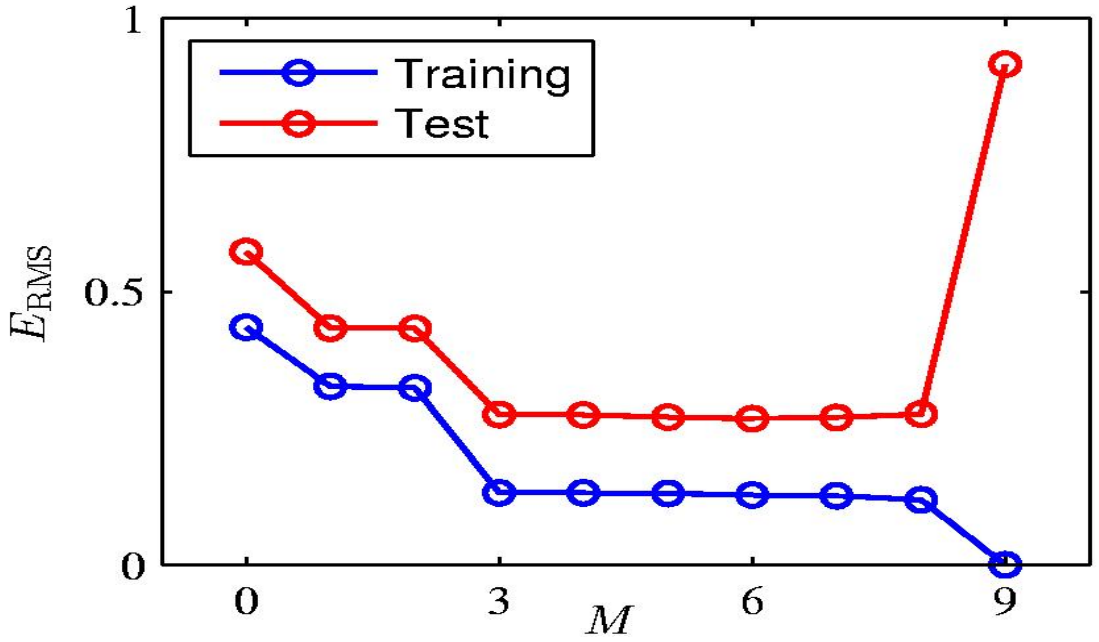
M = 9

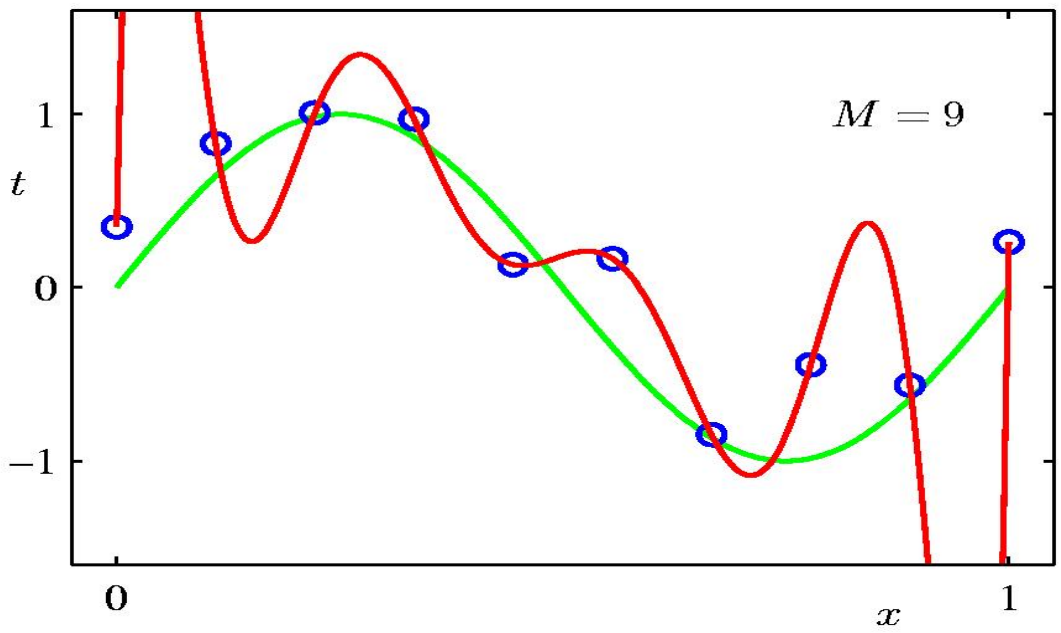


- Fits the training data perfectly but does not perform well on other data points

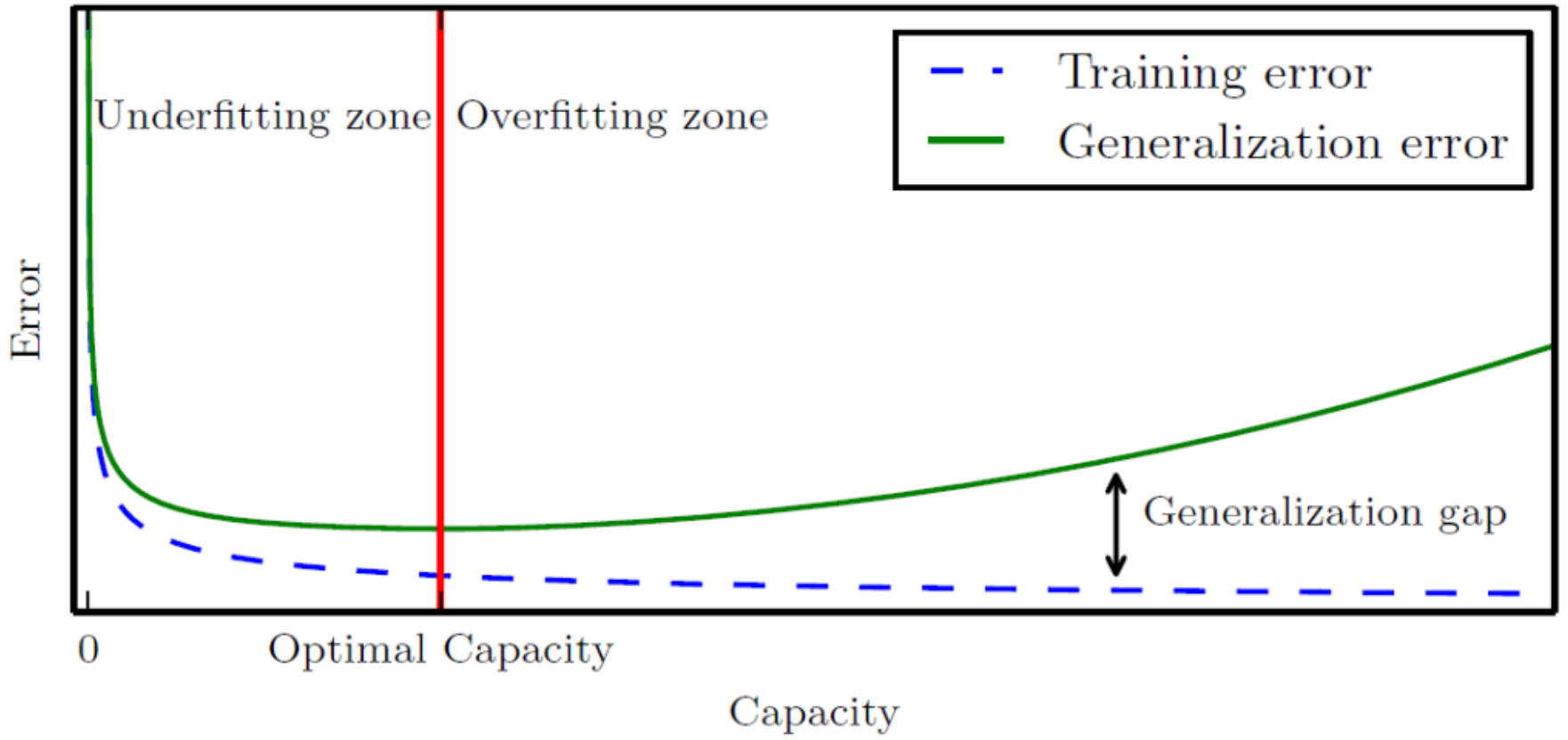
Over-fitting

- Performs very well on the training data while very bad on unseen data (test data)



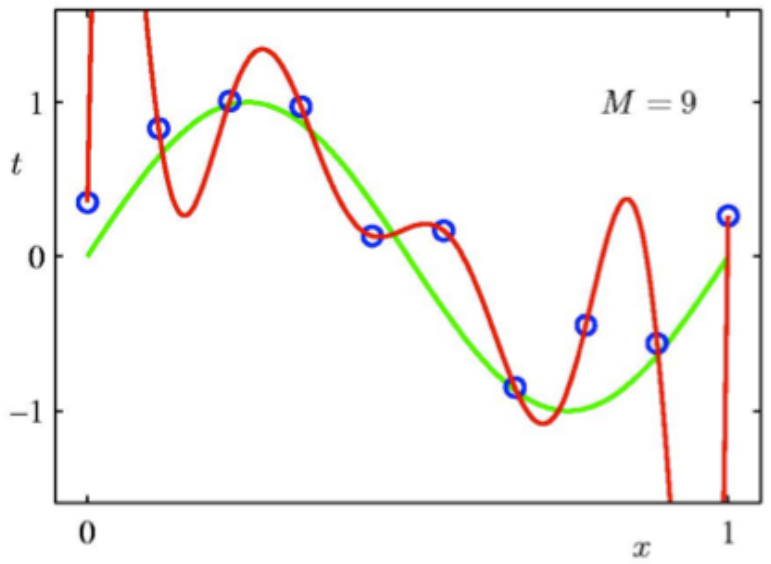


Generalization v.s. Capacity



Polynomial Coefficients

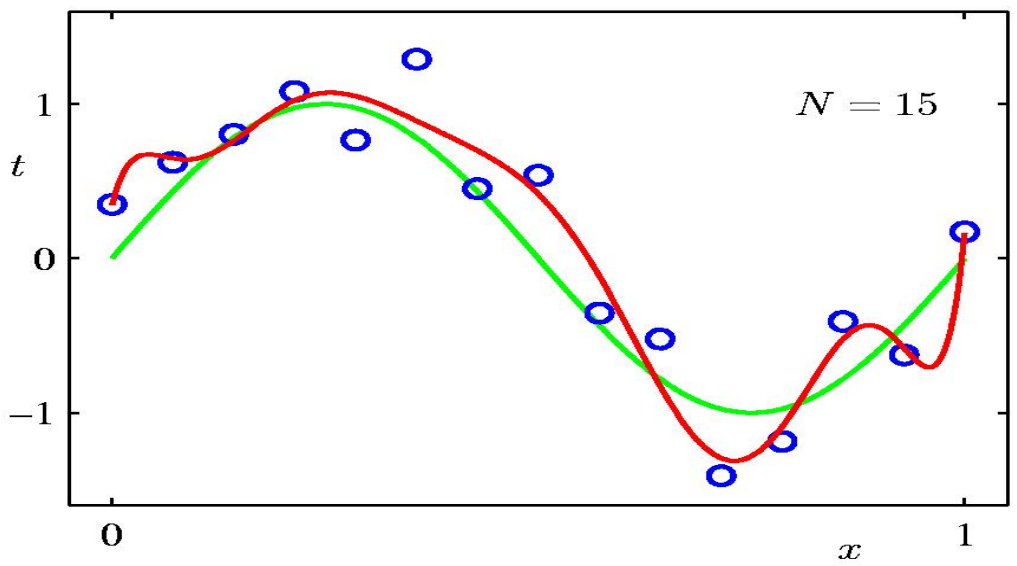
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

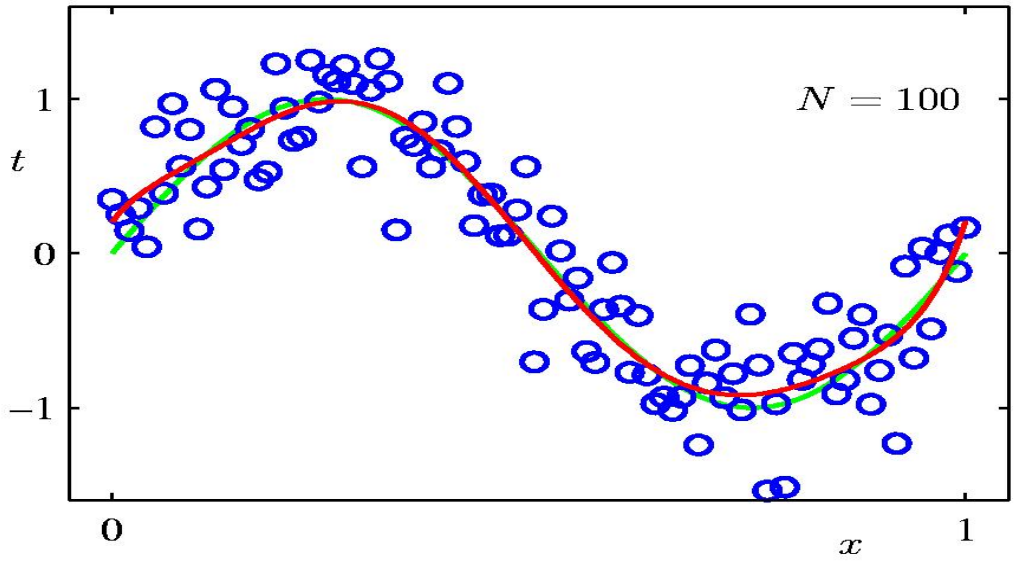


- As M increases, the magnitude of coefficients becomes larger
- M=9, the model overfits to the data

What if we increase the size of data N?

9th Order Polynomial





- The issue of overfitting reduces if we increase the data size N
  - This is one of the reasons why machine learning/deep learning works now

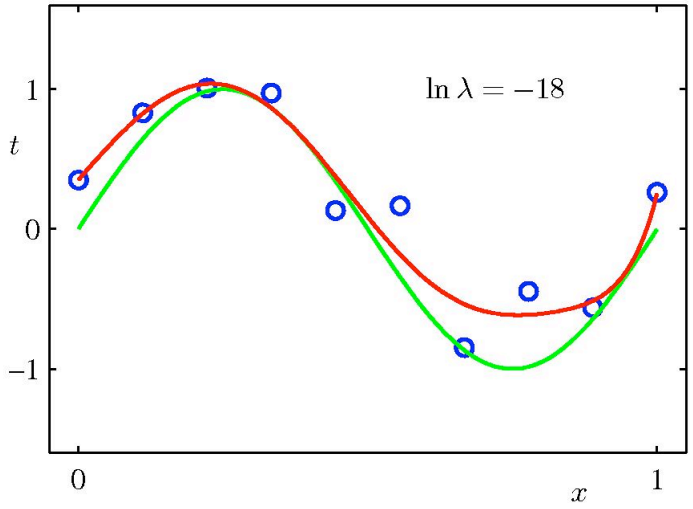
Outline: Linear Regression

- What is Linear Regression
- Model Capacity, Overfitting, and Undefitting
- Regularization
- Cross Validation
- Decision Theory

Regularization

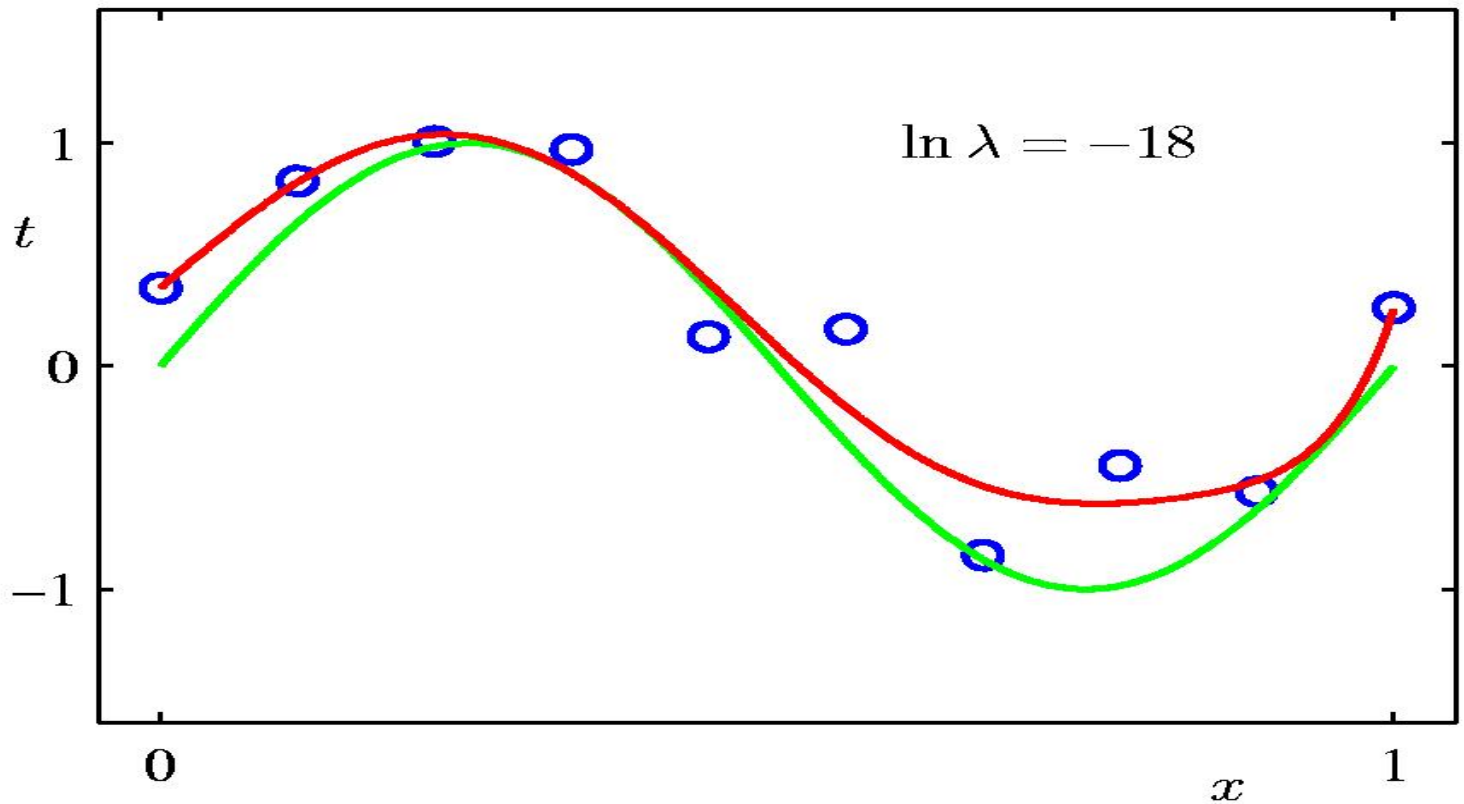
- Penalize large coefficient values

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

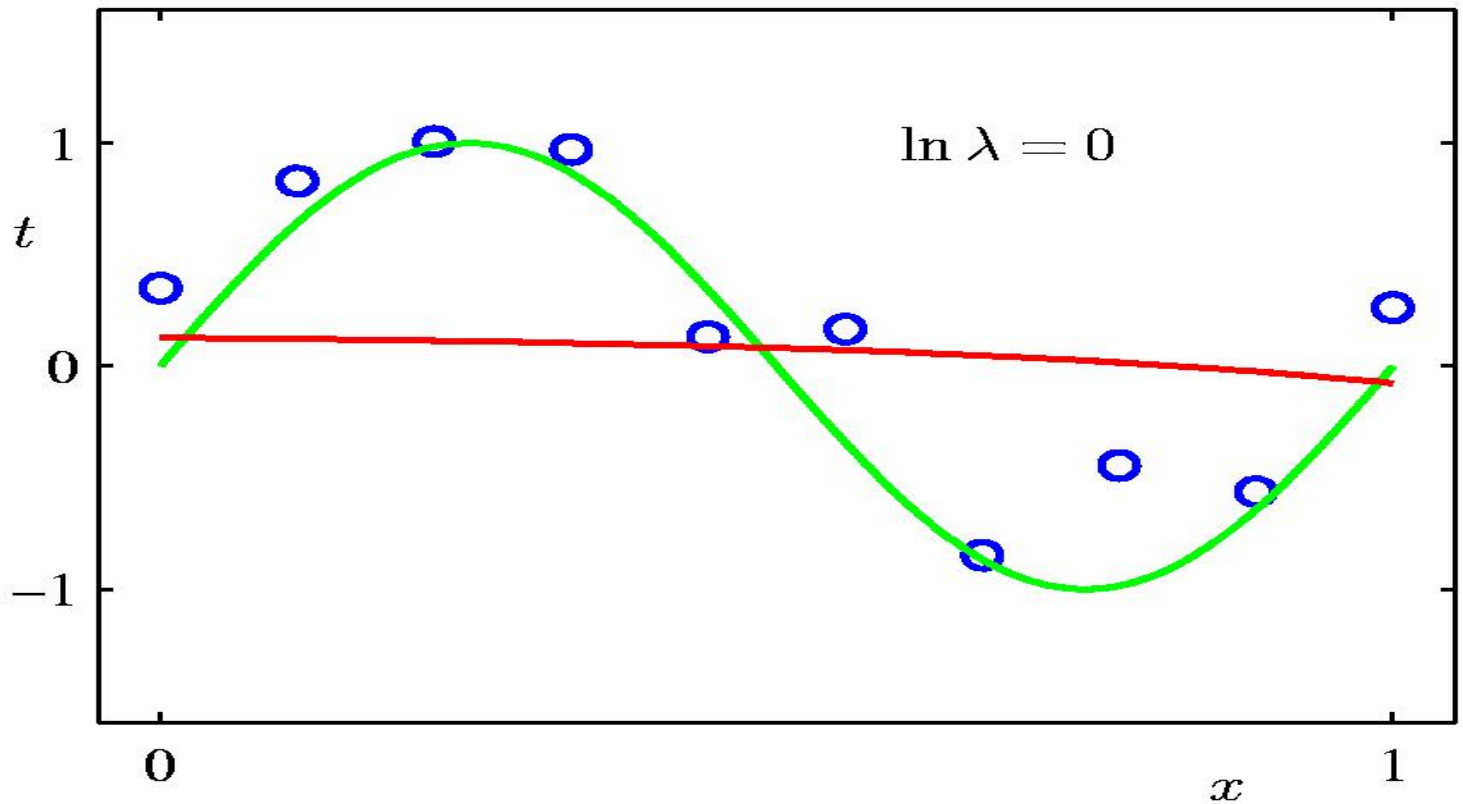


- Tend to choose models with parameters towards zero
  - Penalize coefficients according to their L2 norm
- Also known as “weight decay” or “ridge regression”

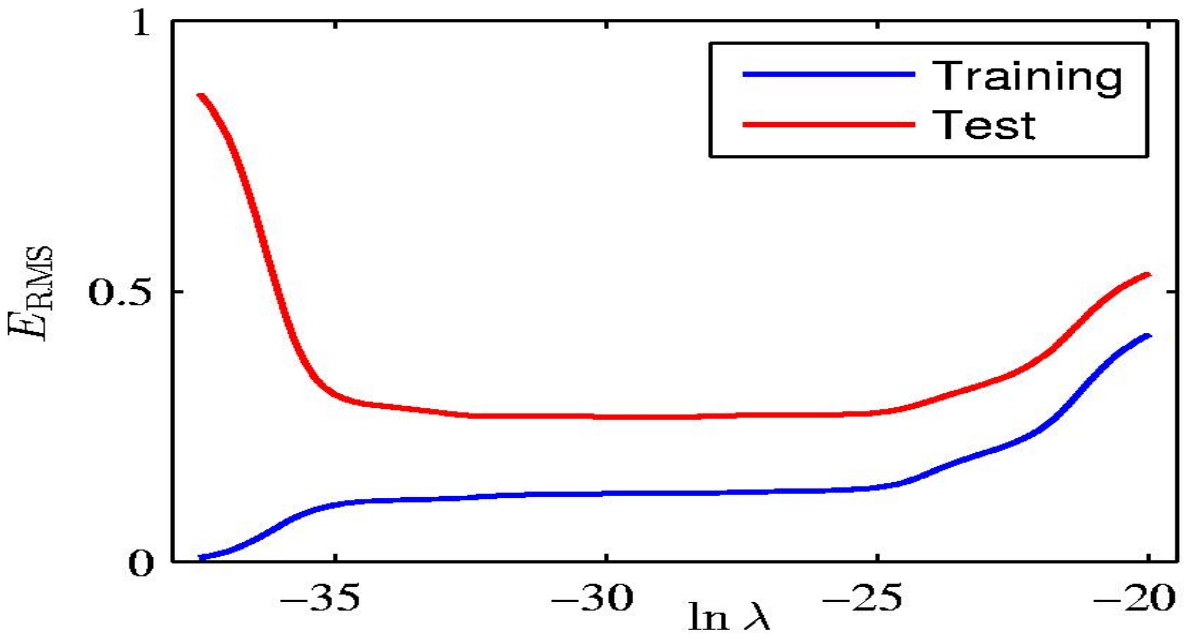
Regularization: $\ln \lambda = -18$



Regularization: $\ln \lambda = 0$



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



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

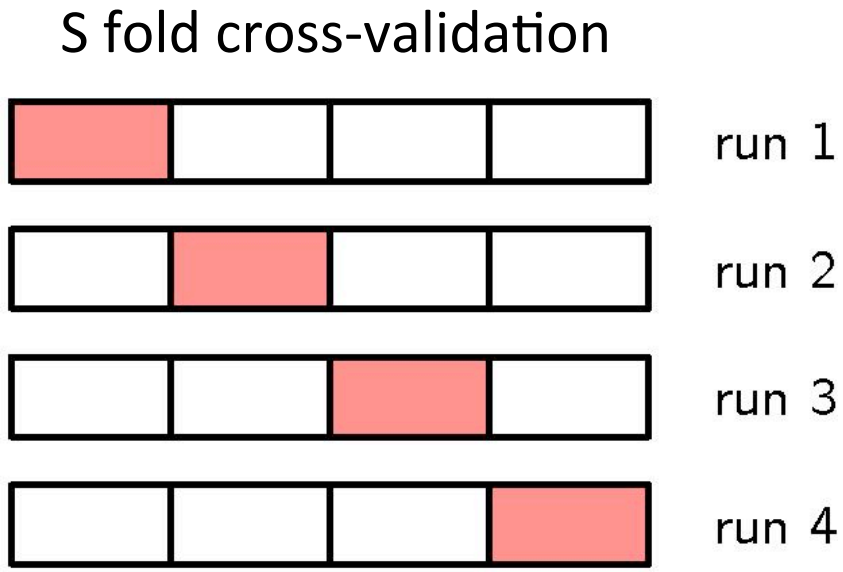
How to choose λ ?

Outline: Linear Regression

- What is Linear Regression
- Model Capacity, Overfitting, and Undefitting
- Regularization
- Cross Validation
- Decision Theory

Cross Validation

- Divide the data set into three subsets
  - Training: used to learn the model parameters
  - Validation: used to select the model,hyper-parameters (e.g., regularization)
  - Test: evaluate the performance of the models
- K-fold cross validation
  - Use as much training data as possible

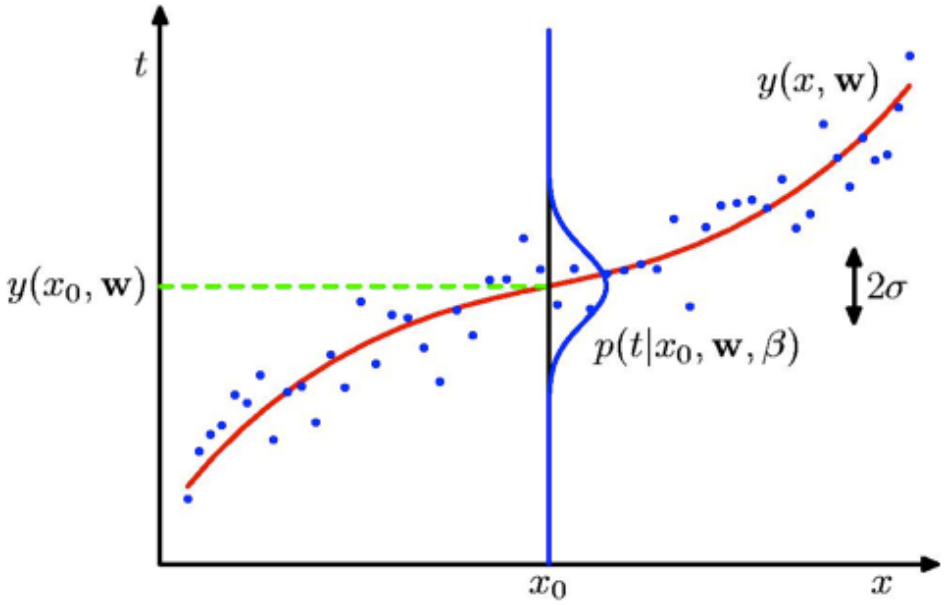


Outline: Linear Regression

- What is Linear Regression
- Model Capacity, Overfitting, and Undefitting
- Regularization
- Cross Validation
- Decision Theory

Probabilistic Perspective of Linear Regression

- Now the objective is modeled as error minimization
- A statistical view: $t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon,$
- ϵ is a random vector having Gaussian distribution with zero mean, and independent with $\boldsymbol{x}$.



$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t|y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}),$$

β is a precision parameter, corresponding to the inverse variance

Maximum Likelihood

- Calculating the Data Likelihood

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n | y(x_n, \mathbf{w}), \beta^{-1}\right)$$

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \underbrace{-\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2}_{\beta E(\mathbf{w})} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

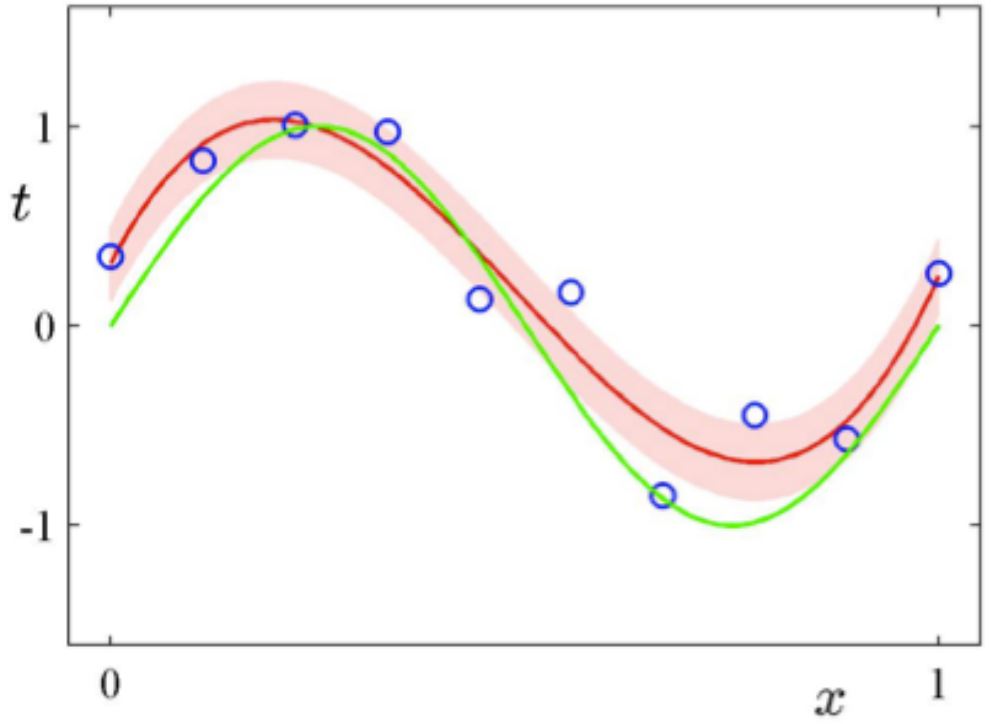
- Maximum data likelihood w.r.t. $\boldsymbol{w}$ = minimizing the sum of squared error
- Maximum data likelihood w.r.t. β:

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n} (y(\mathbf{x}_n, \mathbf{w}_{ML}) - t_n)^2$$

Predictive Distribution

- Once the optimum $\boldsymbol{w}$ and β are found, the predictions for new value $\boldsymbol{x}$ can be calculated as:

$$p(t|\mathbf{x}, \mathbf{w}_{ML}, \beta_{ML}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}_{ML}), \beta_{ML}^{-1})$$



Maximum A Posterior (MAP)-Bayes View

- Bayes view: every parameter is a random variable
- The prior distribution of the parameter

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\}$$

- The posterior distribution of the parameter: $p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta)$
- According to Bayes Theorem:

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

$$\beta\widetilde{E}(\mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Determine $\mathbf{w}_{\mathrm{MAP}}$ by minimizing regularized sum-of-squares error: $\widetilde{E}(\mathbf{w})$

Statistical Decision Theory

• Suppose we have a real-valued input vector **x** and a corresponding target (output) value t with joint probability distribution: $p(\mathbf{x}, t)$.

• Our goal is predict target t given a new value for **x**:
  - for regression: t is a real-valued continuous target.
  - for classification: t a categorical variable representing class labels.

The joint probability distribution $p(\mathbf{x}, t)$ provides a complete summary of uncertainties associated with these random variables.

Determining $p(\mathbf{x}, t)$ from training data is known as the inference problem.

Example: Classification

- Medical image classification: whether the patient has cancer or not

C_1: Cancer present

C_2: Cancer absent

x -- set of pixel intensities

- t as a binary variation: t=0 (with cancer), t=1 (without cancer)

Inference Problem: Determine the joint distribution $p(\mathbf{x}, \mathcal{C}_k)$ or equivalently $p(\mathbf{x}, t)$. However, in the end, we must make a decision of whether to give treatment to the patient or not.

Minimum Expected Loss

- Loss function: measure of the loss incurred by taking any of the available decisions:
  - E.g., suppose x belong to class k while we assign x to class j, this incur loss L_{kj}(an element of a loss matrix)
- Example: medical image classification

Truth \ Decision	cancer	normal
cancer	0	1000
normal	1	0

Expected Loss:

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}$$

Goal is to choose decision regions $\mathcal{R}_j$ as to minimize expected loss.

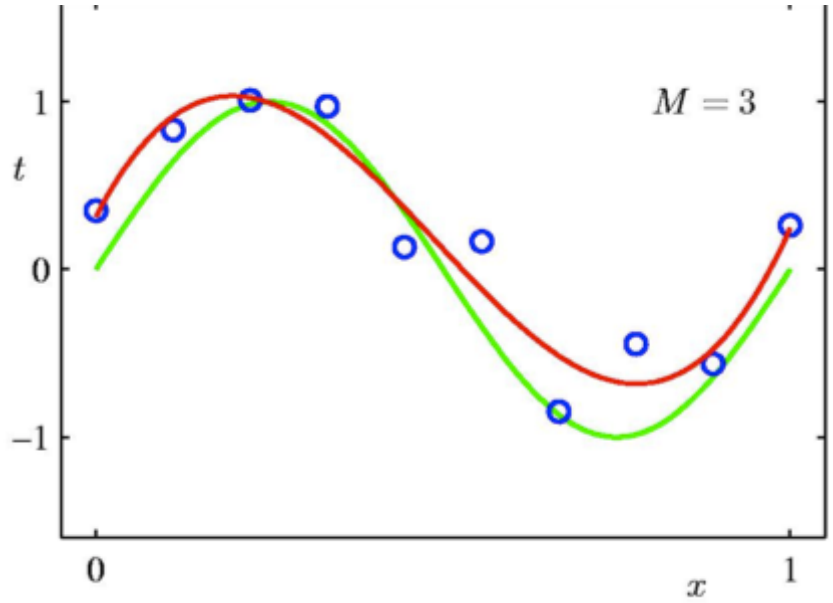
Example: Regression

- x: a real-valued input vector variable, t: a real-valued output (target) variable, with the joint distribution p(x,t)
- Decision: find an estimate y(x) of t for each input x
- Define the loss function: $L\big(t, y(x)\big)$.
- The average/expected loss

$$\mathbb{E}[L] = \int \int L(t, y(\mathbf{x}))p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t.$$

- Squared loss:

$$\mathbb{E}[L] = \int \int \big(t - y(\mathbf{x})\big)^2 p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t.$$



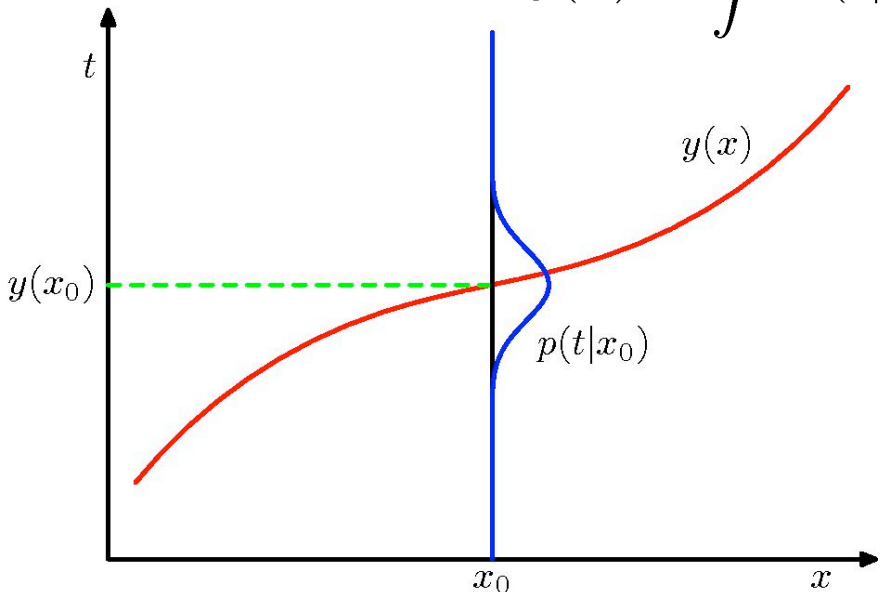
Square Loss Function

- Square loss function:

$$\mathbb{E}[L] = \int \int \big(t - y(\mathbf{x})\big)^2 p(\mathbf{x}, t) \mathrm{d}\mathbf{x} \mathrm{d}t.$$

- Goal: find y(x) to minimize the above objective function
- The optimal solution is the conditional average:

$$y(\mathbf{x}) = \int t p(t|\mathbf{x}) \mathrm{d}t = \mathbb{E}[t|\mathbf{x}]$$



The regression function y(**x**) that minimizes the expected squared loss is given by the mean of the conditional distribution $p(t|\mathbf{x})$.

Generative v.s. Discriminative Model

- Generative model:
  - Model the joint probability $p(t, \mathbf{x}) = p(\mathbf{x}|t)p(t)$
  - Make decision according to Bayes theorem

$$p(t|\mathbf{x}) = \frac{p(\mathbf{x}|t)p(t)}{p(\mathbf{x})}$$

- Discriminative model:
  - Directly model the conditional probability $p(t|\mathbf{x})$

References

- Chapter 5, Deep Learning Book.