Machine Learning

Part 1: Introduction to Pattern Recognition and Machine Learning

Zengchang Qin (PhD)

What is Machine Learning

1) Function : $x \rightarrow f(x)$

Given $\boxed{f(x) = 2x + 1}$

Input	x	$x=1$	$x=2$	...	$x=10$
output	$f(x)$	$f(x)=3$	$f(x)=5$	...	$f(x)=21$

2) If we are only given :

x	$x=1$	$x=2$	...	$x=10$
$f(x)$	$f(x)=3$	$f(x)=5$	...	$f(x)=21$

what is $f(x) = ?$

3) The world is complicated, usually, the data is like the following:

x	$x=1$	$x=2$	...	$x=10$
$f(x)$	$f(x)=3.2$	$f(x)=4.9$	...	$f(x)=20.9$

Q: what is $f(x) = ?$

What is Machine Learning

4) Now, we introduce the Vector.

E.g. $X = (x_1, x_2)$, $f(x) = 2x_1 + 3x_2 - 1$

If we are given the following table:

$X = (x_1, x_2)$	$(1, 1)$	$(1, 2)$	$\cdots$	$(4, 3)$
$f(x) = f(x_1, x_2)$	$f(x) = 4.1$	$f(x) = 6.9$	$\cdots$	$f(x) = 15.8$

Q: what is $f(x) = f(x_1, x_2) = ?$

5) Given an image:  It can be represented by a vector $(x_1, x_2 \cdots x_n)$

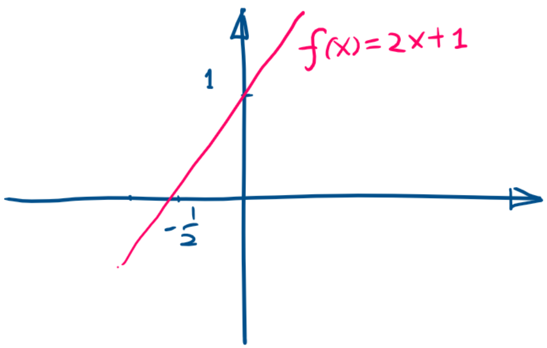
we need to learn a function

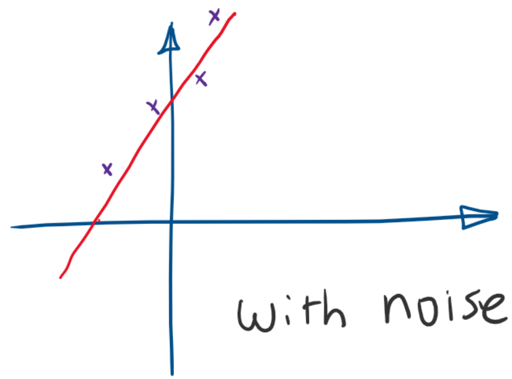
$f($  $) = f(x_1, x_2 \cdots x_n) =$ Dog

$f($  $) = f(x_1, x_2 \cdots x_n) =$ Cat

What is Machine Learning

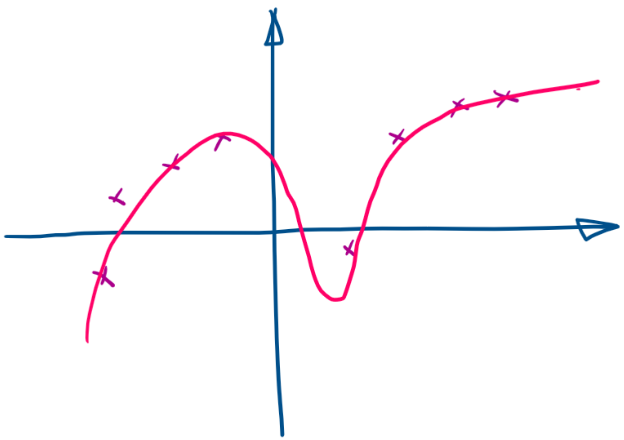
6) Some functions are easy, Some are difficult.



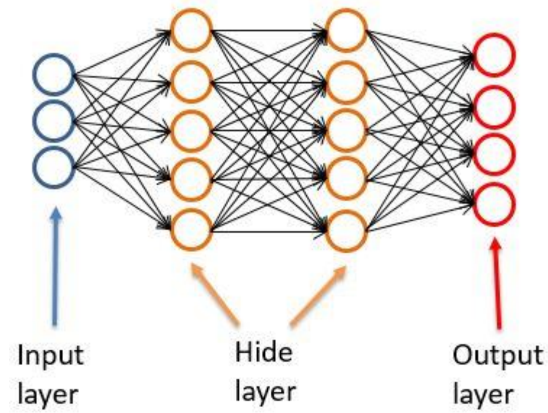


For complicated case like map a vector of an image, we may not find an existing function. we may need a model which can approximate any functions.
Artificial Neural Network is such a model.

What is Machine Learning



For Complicated cases
we need to use a
NN to approximate
the underlying relations



$h_\Theta(x) \in \mathbb{R}^4$

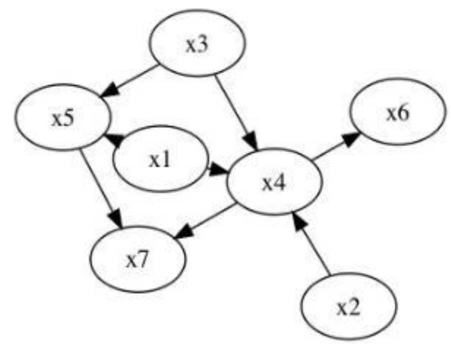
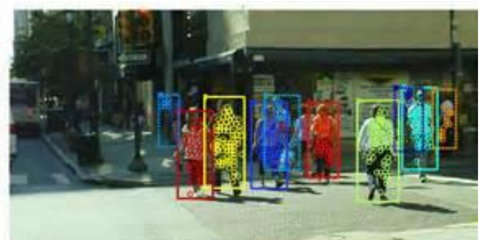
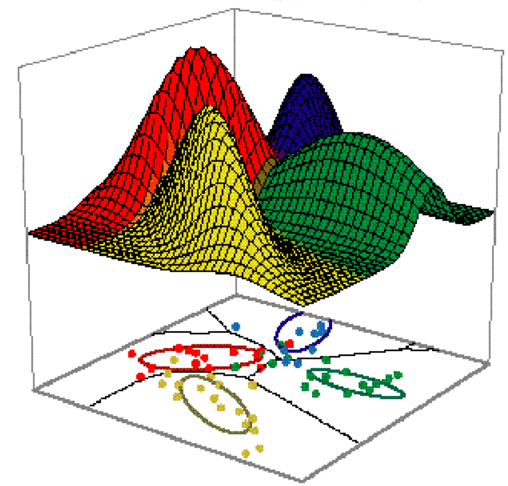


图2

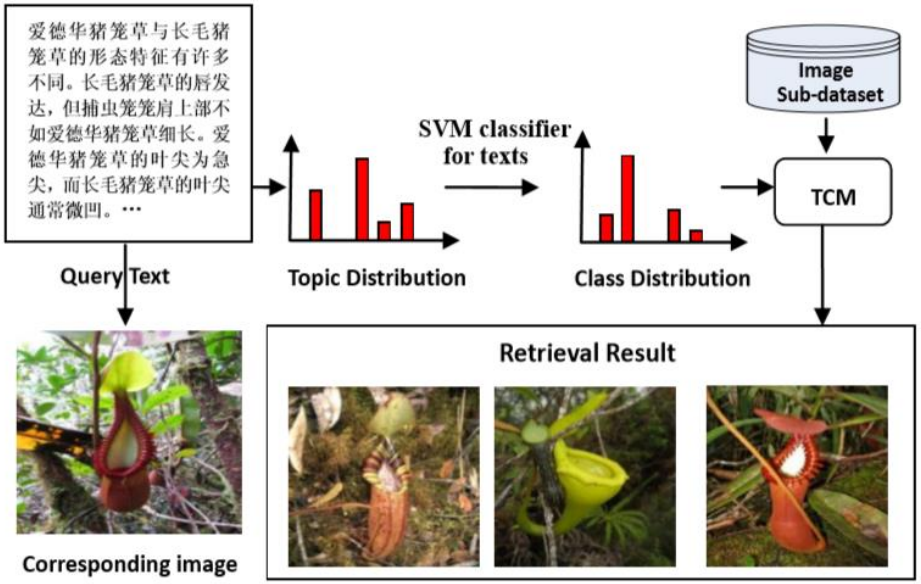
Machine Learning Applications

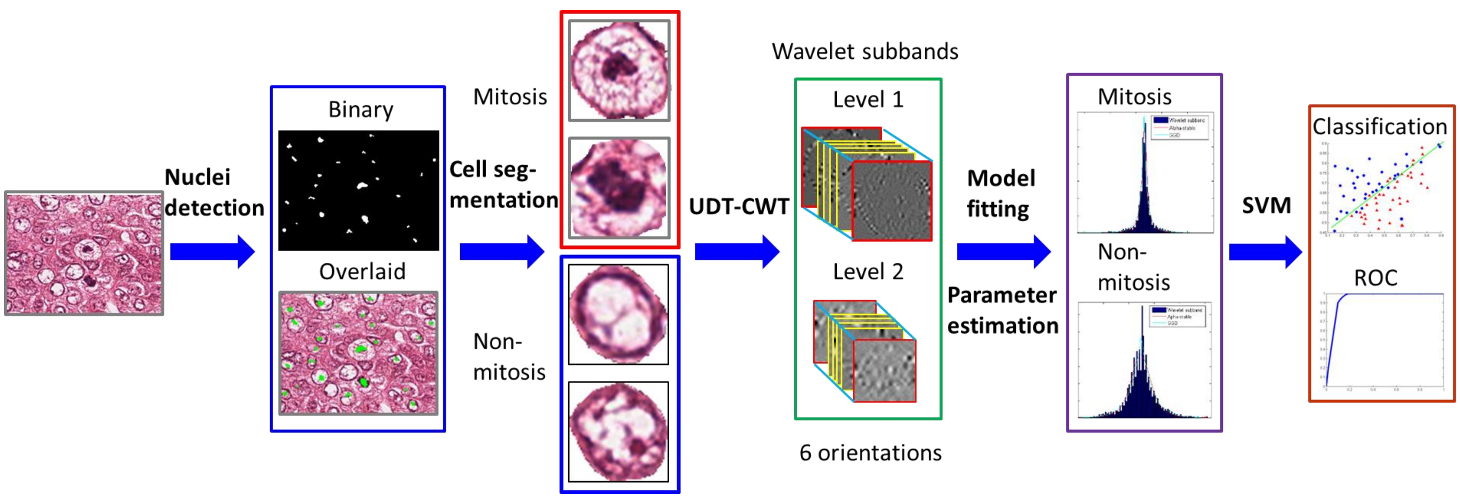
- Artificial Intelligence
- Pattern Recognition
- Machine Learning
- Data Mining (Big Data)
- Statistical Learning
- Deep Learning
- Natural Language Processing
- Information Retrieval
- Computer Vision
- Image Processing
- Speech Recognition
- Biometrics
- Bioinformatics



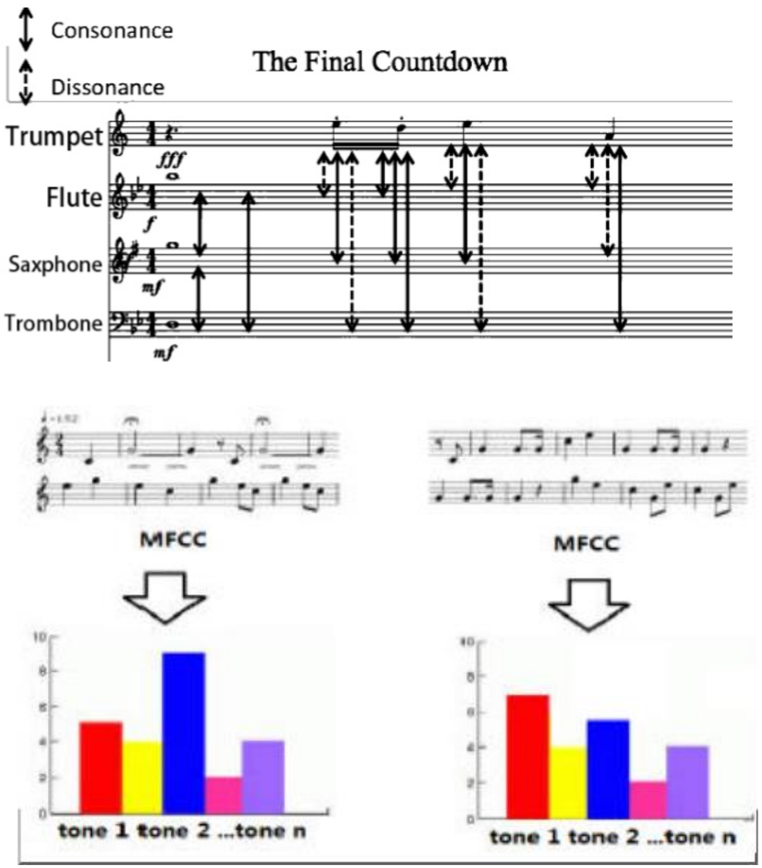


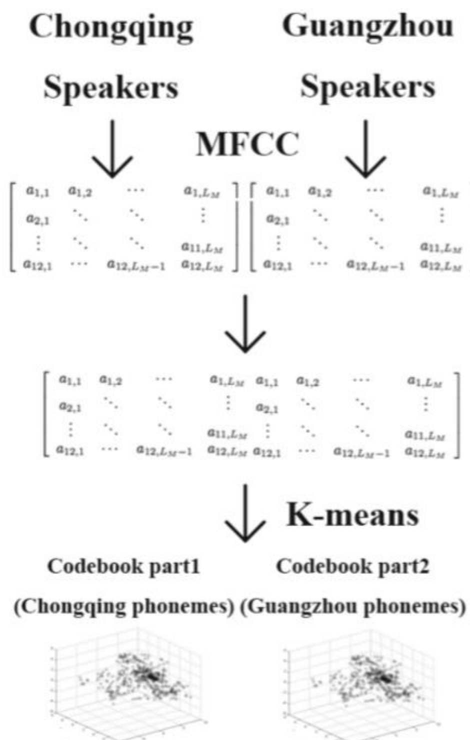
Machine Learning Applications

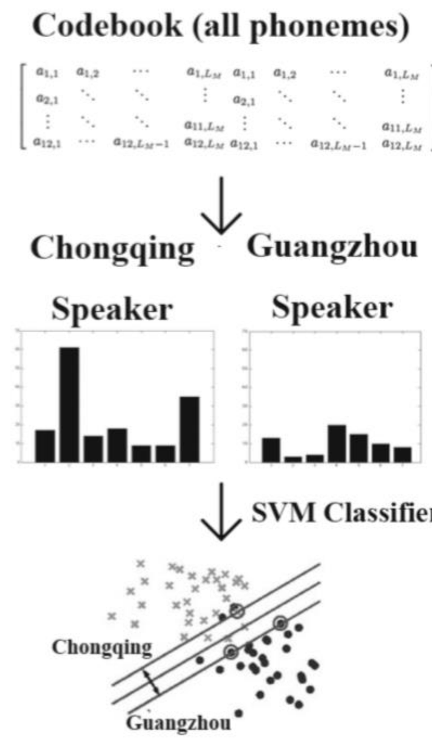




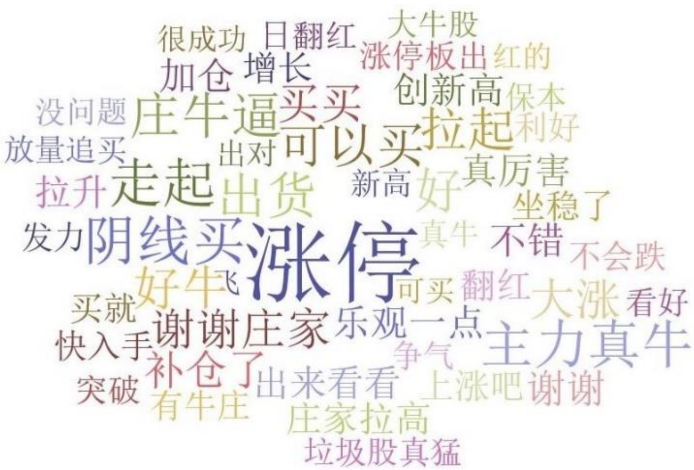
Machine Learning Applications

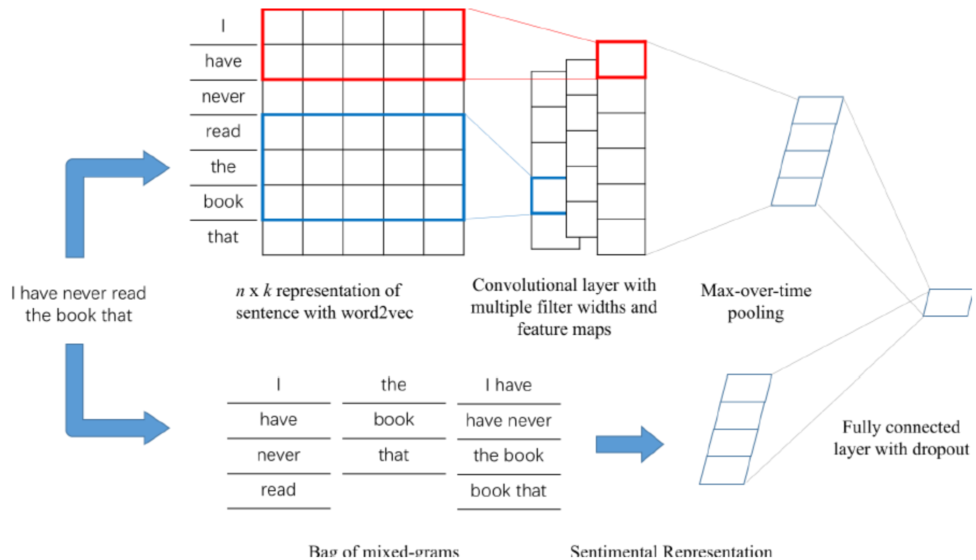


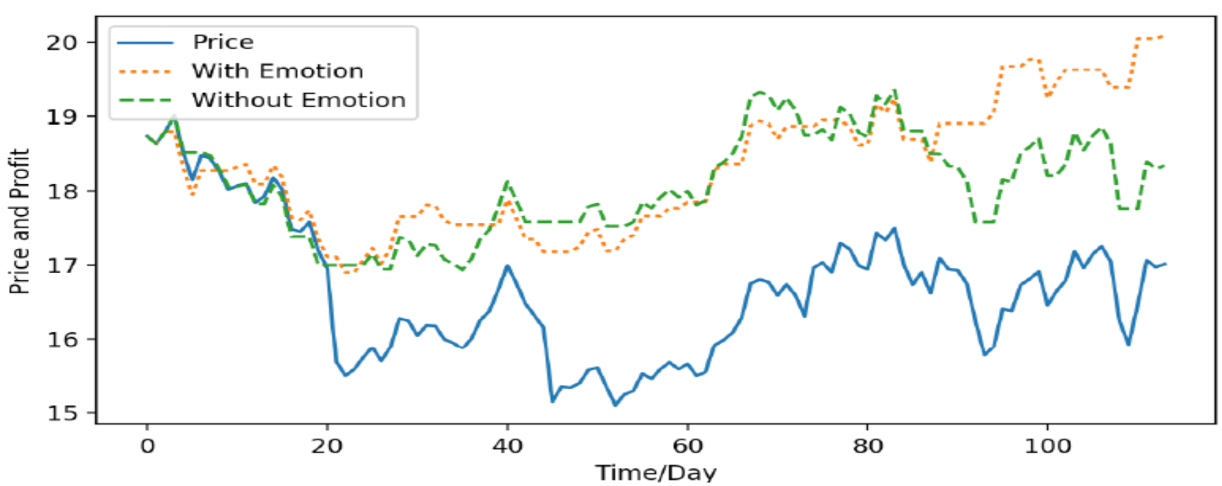




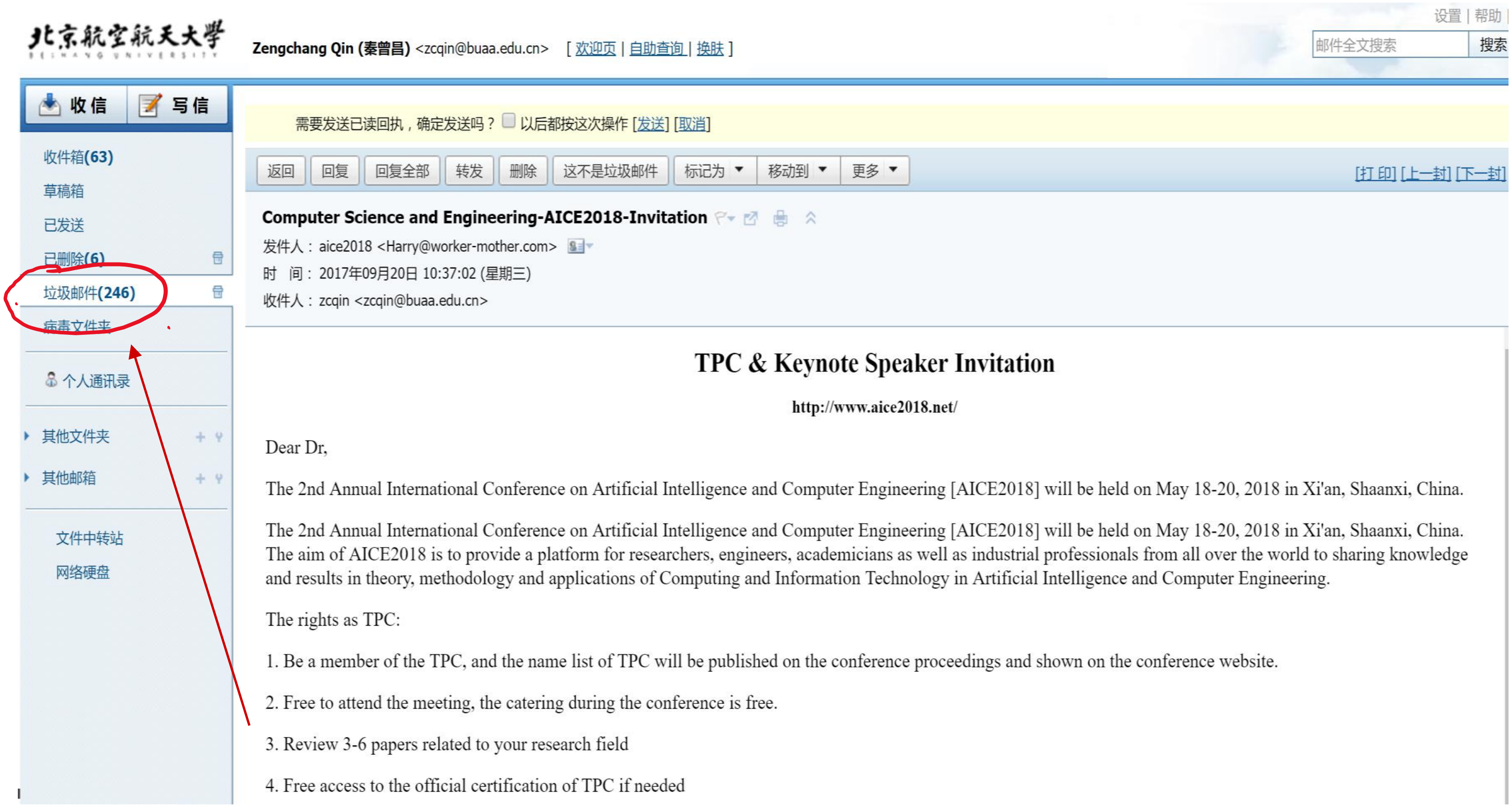
Machine Learning Applications



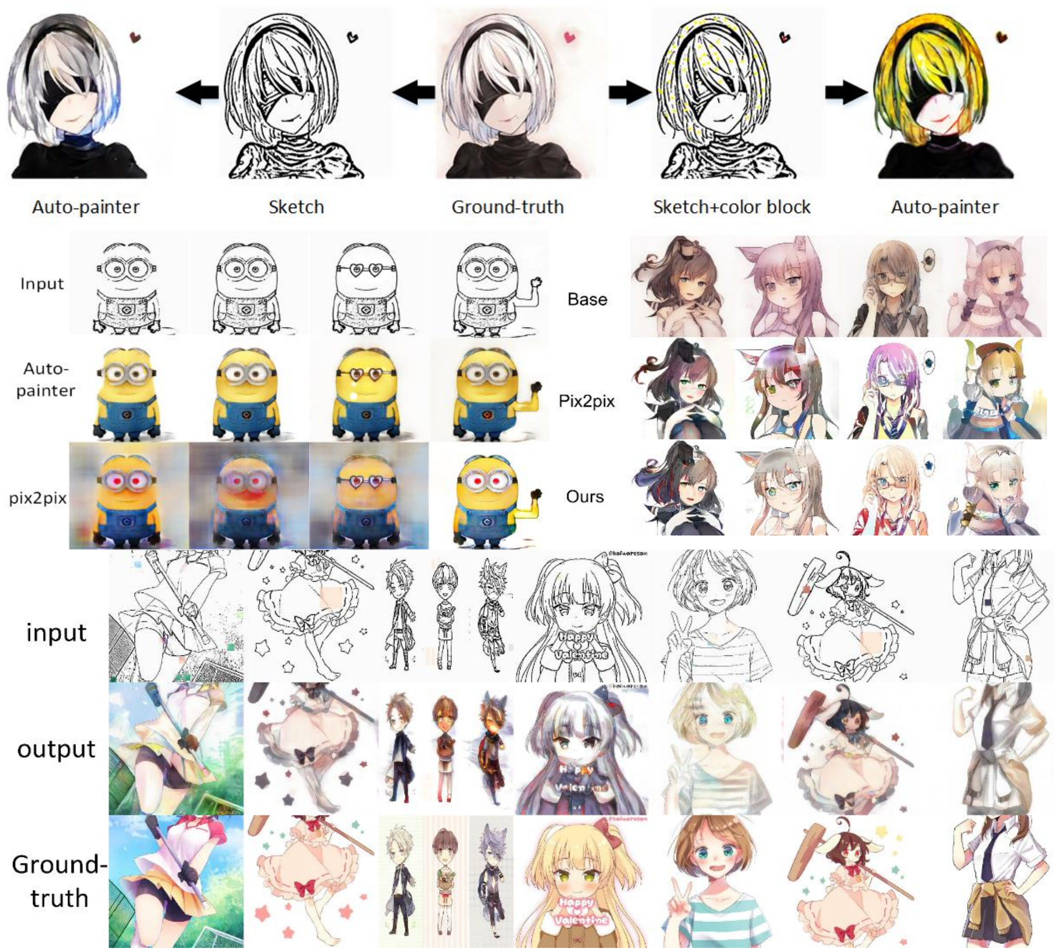


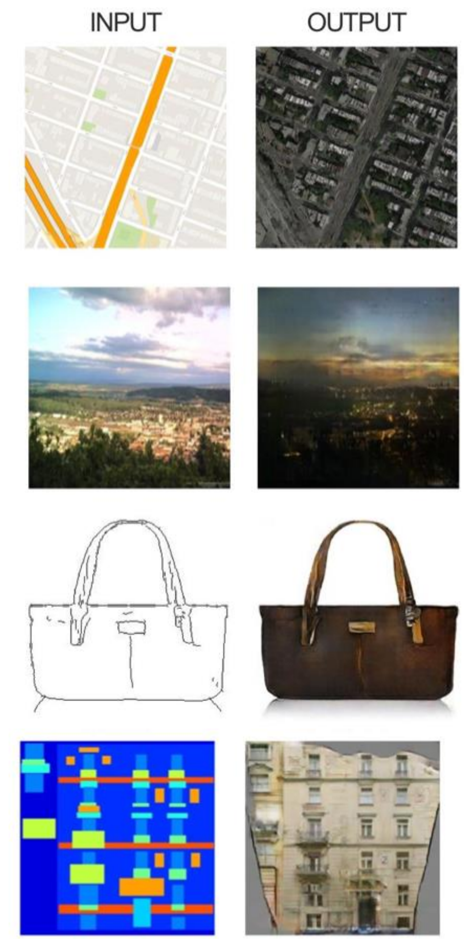


Machine Learning Applications

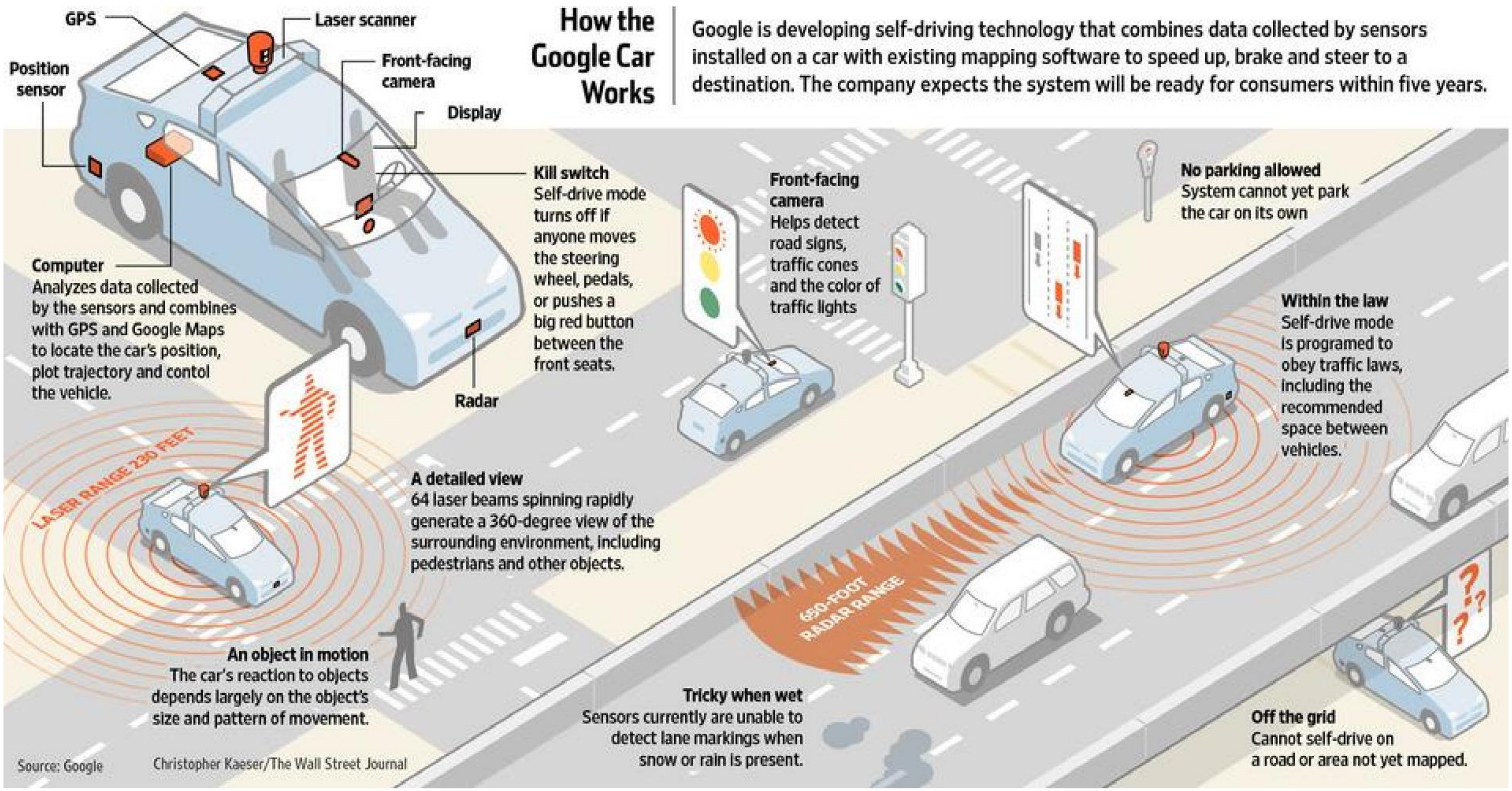


Machine Learning Applications

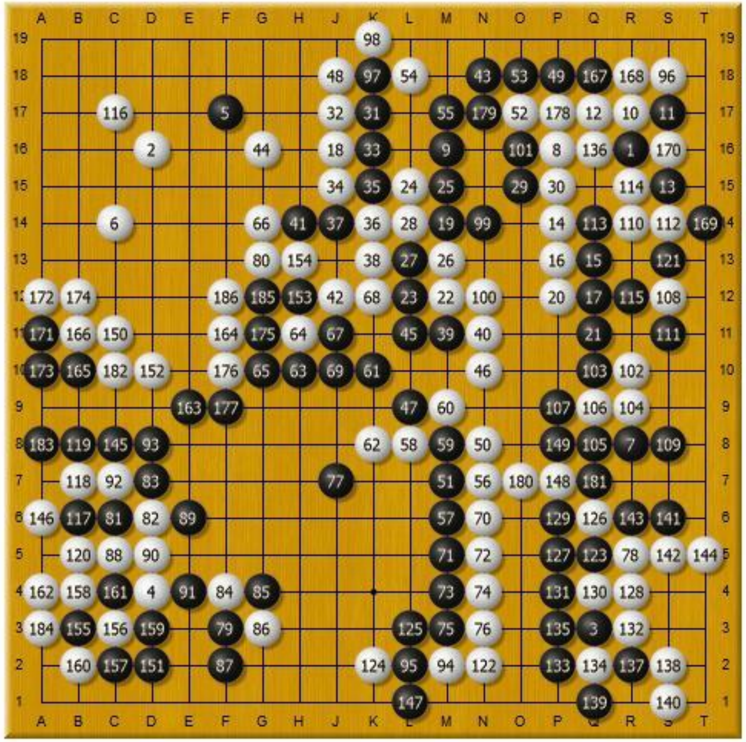




Machine Learning Applications



Machine Learning Applications



Science of Algorithm

■ If machine learning is a science, it is a science of algorithms.   - Pat Langley [1]

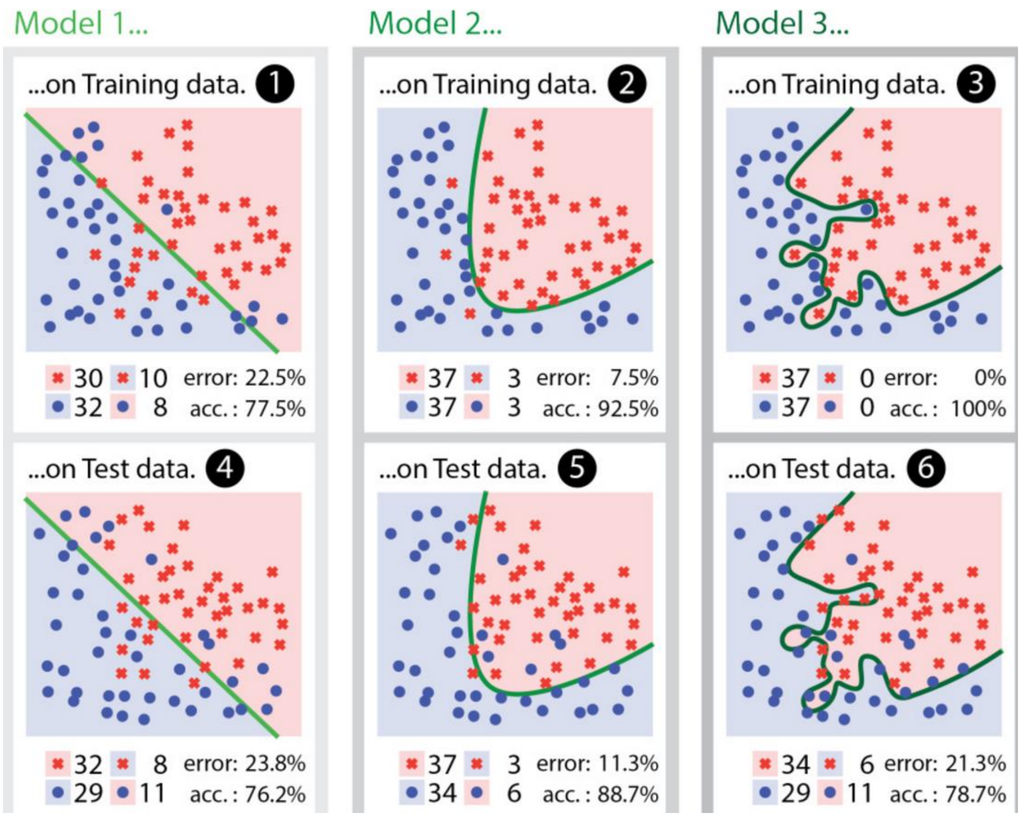
■ The central idea, already proposed by Simon [2] was that the purpose of learning is to improve performance on some class of tasks.

■ The UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) is available to the community by FTP in 1987.

■ Like science, it is about hypothesis testing using evidence.

[1] Langley, P. (1986). Human and machine learning. Machine Learning, 1, 243–248.
[2] Simon, H. A. (1983). Why should machines learn? In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), Machine learning: An artificial intelligence approach. San Mateo, CA: Morgan Kaufmann

Machine Learning from Data is Just the Process of Scientific Discovery

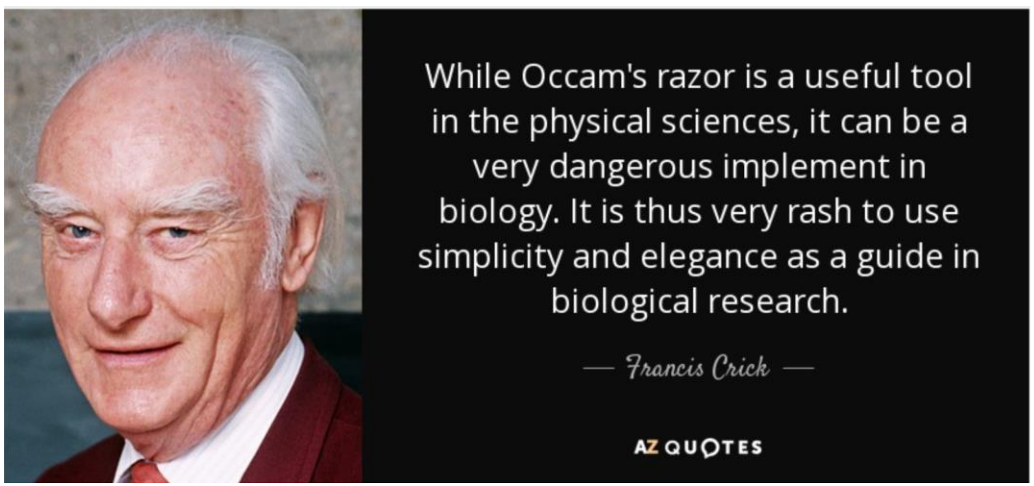


- A tradeoff between generalization and specification. (E.g, there is no two leave are exactly the same, or all greens are leaves.)

"I've narrowed it to two hypotheses: it grew or we shrunk."

Occam's Razor and Overfitting

■ **Occam's razor** (or **Ockham's razor**) is a principle from philosophy. Suppose there exist two explanations for an occurrence. In this case, the simpler one is usually better.



Ockham chooses a razor

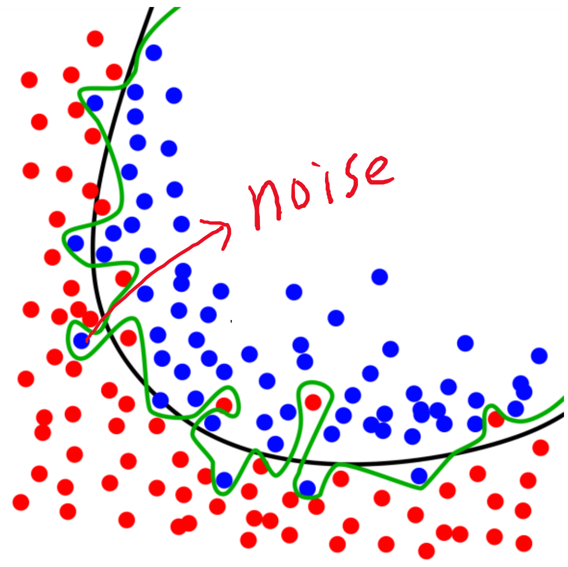
Q: Any other examples of Occam's Razor?

Overfitting

- Let's say you attend a symphony and want to get the clearest, most faithful sound possible. So you buy a super-sensitive microphone and hearing aid to pick up all the sounds in the auditorium.

- **Then you start "overfitting," hearing the noise on top of the symphony.** You hear your neighbors shuffling in their seats, the musicians turning their pages, and even the swishing of the conductor's coat jacket.

- In machine learning, we tends to favor simple models. When two models that make exactly the same predictions, the simpler one is the better. If the model is too complex, we may face the problem of **overfitting**.



Q: Any other examples of overfitting?

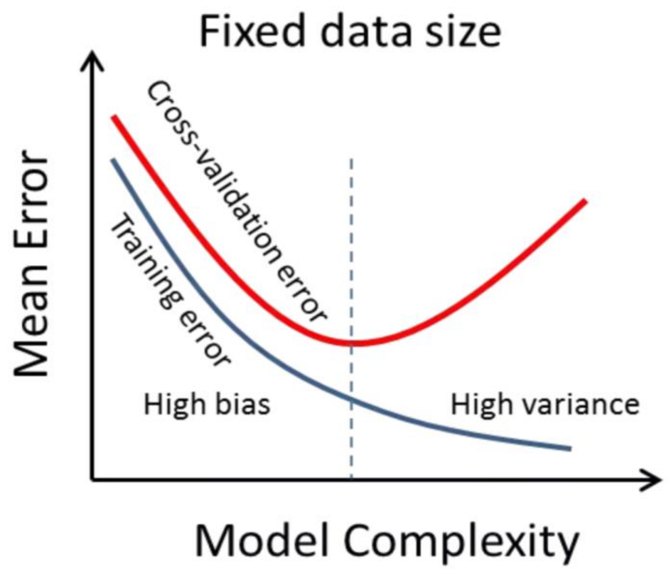
Avoid Overfitting

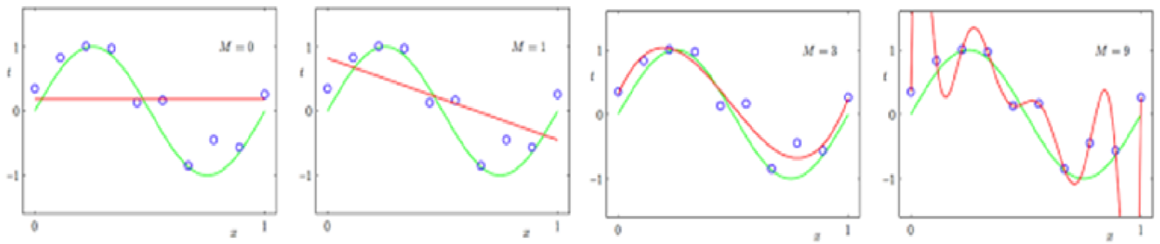
In short, the general strategies are to

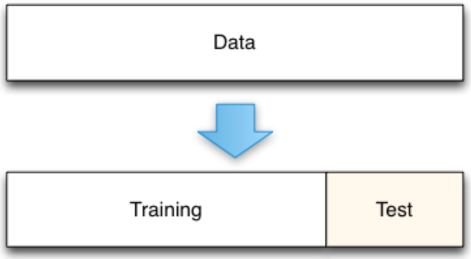
1.Collect more data
2.Use ensemble methods that "average" models
3.Regularization to penalize complexity

$$W = \sum V(f(x_i), t_i) + \lambda\Omega(f)$$

4. N-fold cross-validation



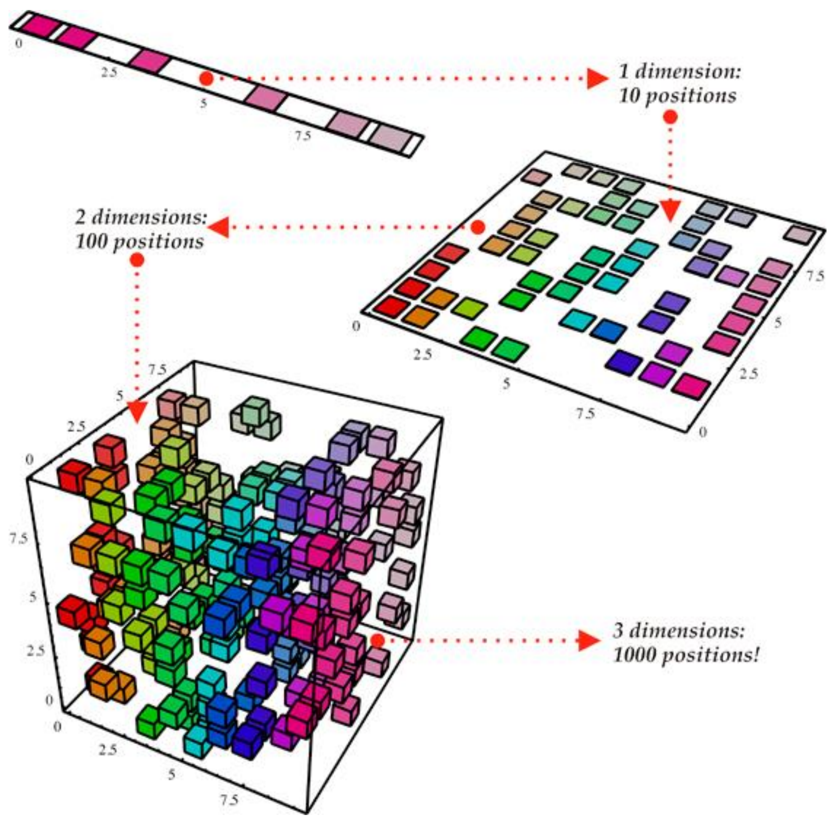




Curse of Dimensionality

Imagine that in order to produce a good prediction, our learner needs to distinguish between 10 different values of each of n variables. Then it may need to distinguish between 10^n different configurations of the input n-dimensional vector. With n easily in the hundreds, thousands or more, this is much more than the number of examples one can hope to gather (or even the number of atoms in the universe) [3].

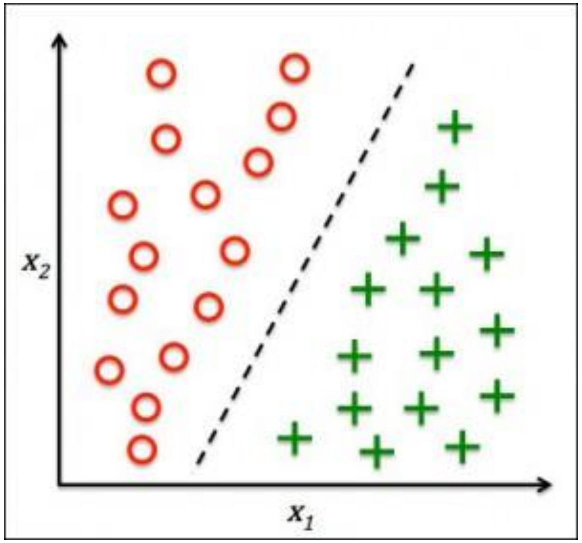
Q: What a high-dimensional sphere looks like?



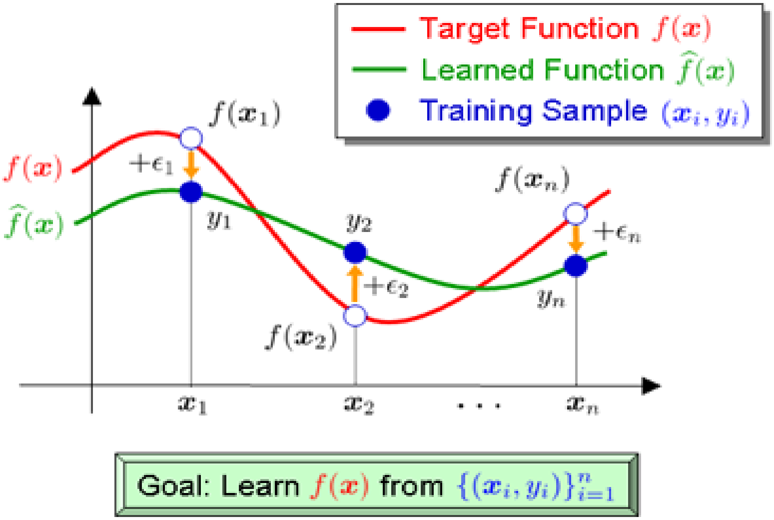
[3] Yoshua Bengio and Yann LeCun, Scaling Learning Algorithms towards AI, Large-Scale Kernel Machines, L. Bottou, O. Chapelle, D. DeCoste, J. Weston (eds) MIT Press, 2007

Supervised Learning

Supervised Learning as classification



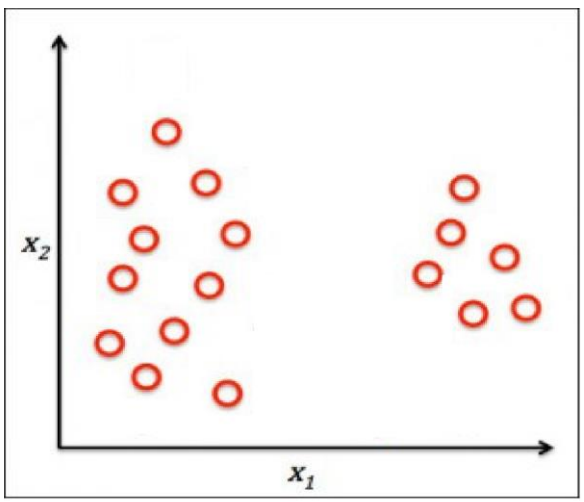
Supervised Learning as Function Approximation

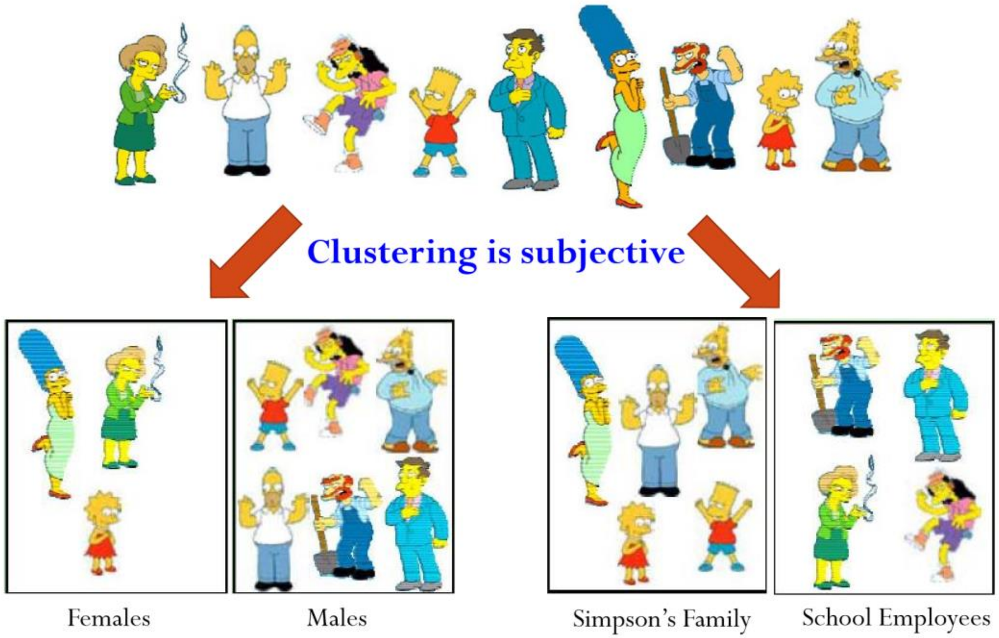


Classification is also a prediction of f(x) -> {0, 1}

Unsupervised Learning (Clustering)

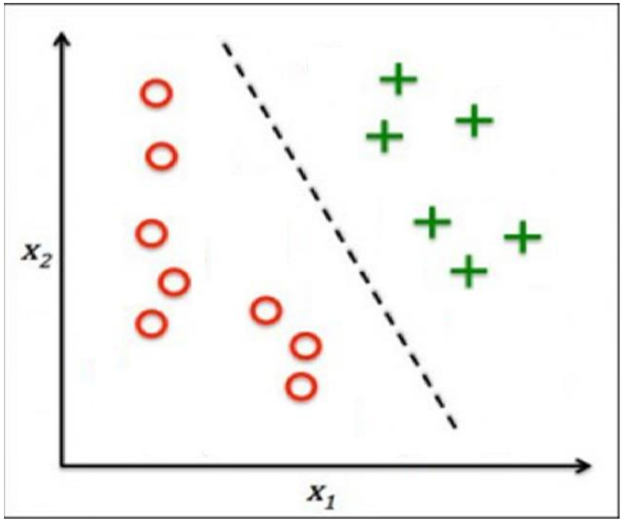
Unsupervised Learning



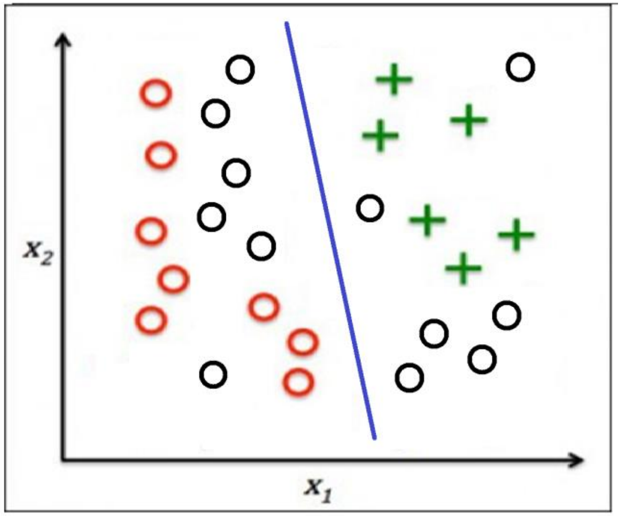


Semi-supervised Learning

All labeled data

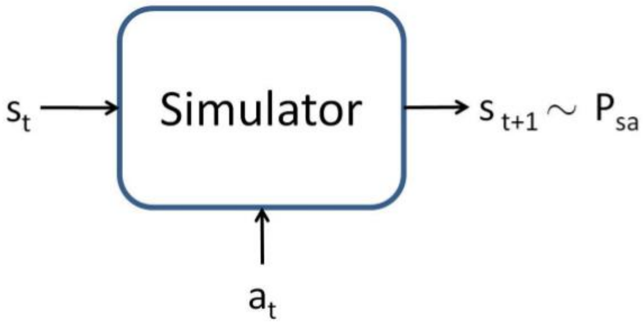


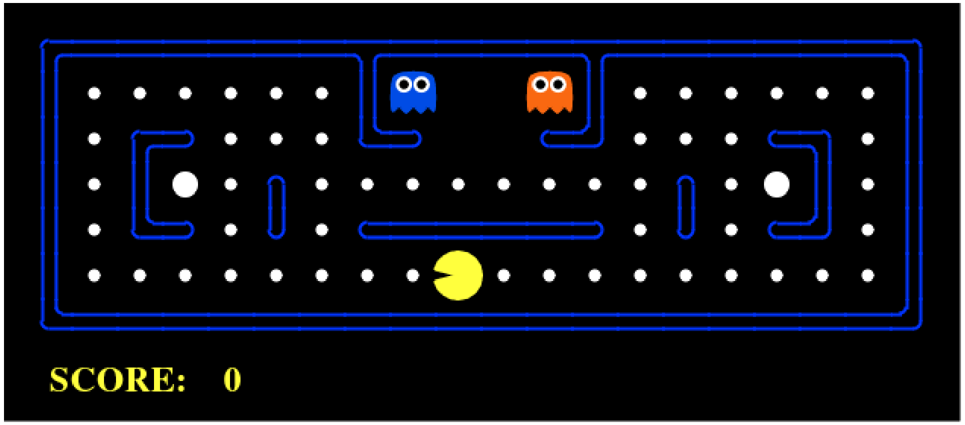
With unlabeled data



Reinforcement Learning

Informally, a simulator is a black-box that takes as input any (continuous-valued) state s_t and action a_t, and outputs a next state s_{t+1} sampled according to the state transition probabilities P_{sa} :

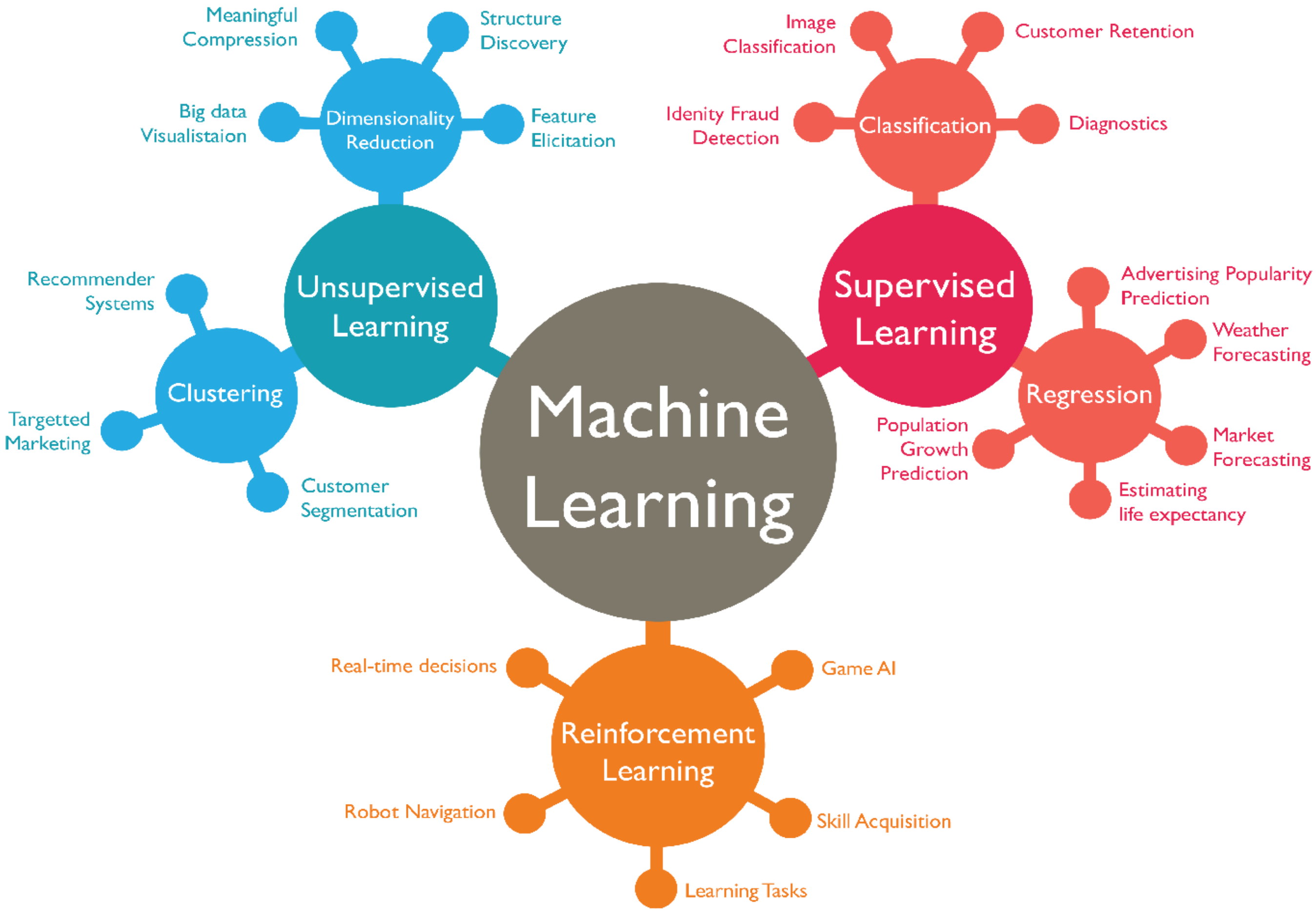




PAC Man with Reinforcement Learning

Q: Advantages and disadvantages of RL?

Paradigm Map of Machine Learning



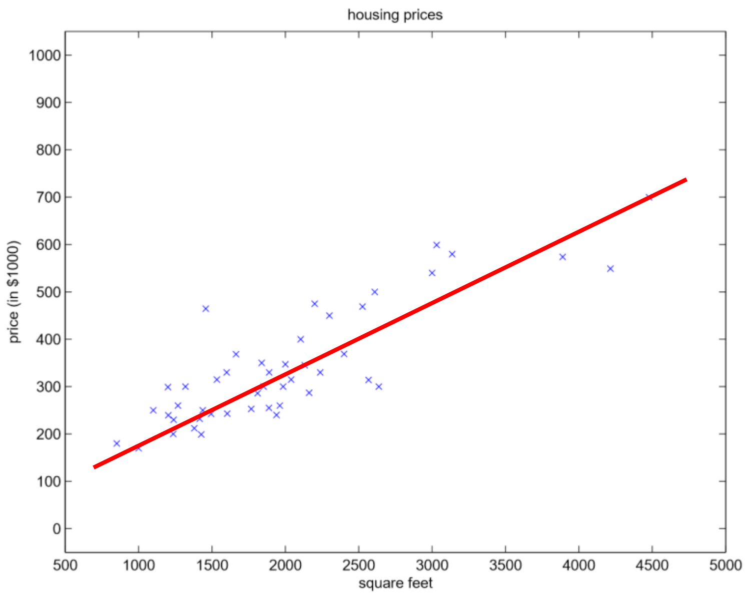
From: https://www.normshield.com/machine-learning-in-cyber-security-domain-1-fundamentals/

Housing Price in Portland

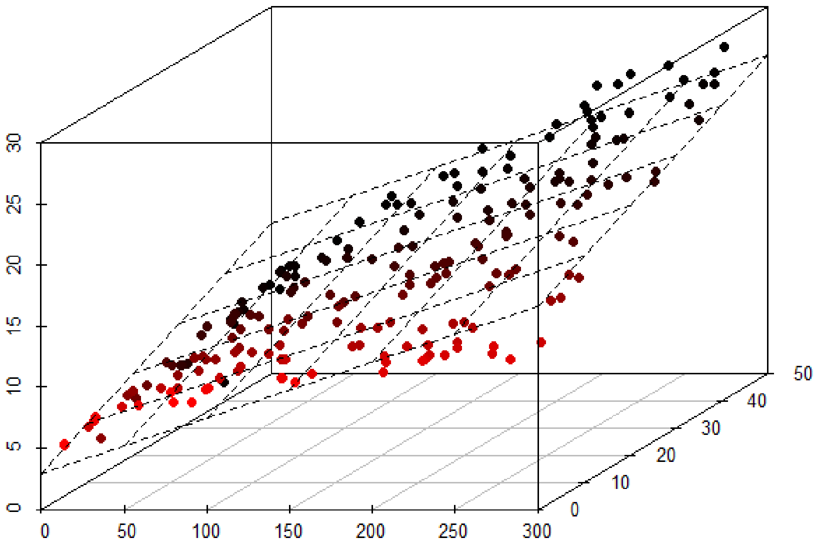
In **Andrew Ng's** Lecture, there is a dataset giving the living areas and prices of 47 houses from Portland, Oregon. We are looking for a function gives the pattern of inputs-outputs.

Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

$$h_\theta(x) = \theta_0 + \theta_1 x$$



Multi-dimensional Linear Regression



From one dimension to 2 dimensional case:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Multi-dimensional Attributes (Features)

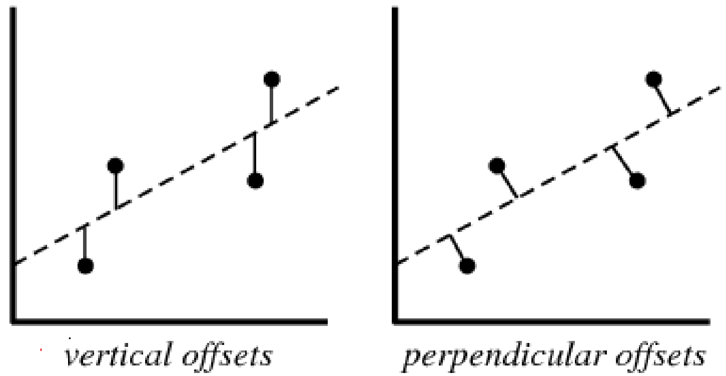
A pair $(x^{(i)}, y^{(i)})$ is called a **training** example, $x \in R^d$ is called the **feature** and y is called the target or label of the example.

To perform **supervised learning**, we must decide how we're going to represent functions/hypotheses h.

Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

Least Square Fitting

A mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets.



$$S = \sum_{i=1}^{n} {r_i}^2 \qquad r_i = y_i - f(x_i, \boldsymbol{\beta}).$$

Solving the least squares problem:

$$\frac{\partial S}{\partial \beta_j} = 2\sum_i r_i \frac{\partial r_i}{\partial \beta_j} = 0,\ j = 1, \ldots, m,$$

$$= -2\sum_i r_i \frac{\partial f(x_i, \boldsymbol{\beta})}{\partial \beta_j} = 0,\ j = 1, \ldots, m.$$

since $r_i = y_i - f(x_i, \boldsymbol{\beta})$

Residuals are the vertical distances between the data points and the corresponding predicted values.

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{y}.$$