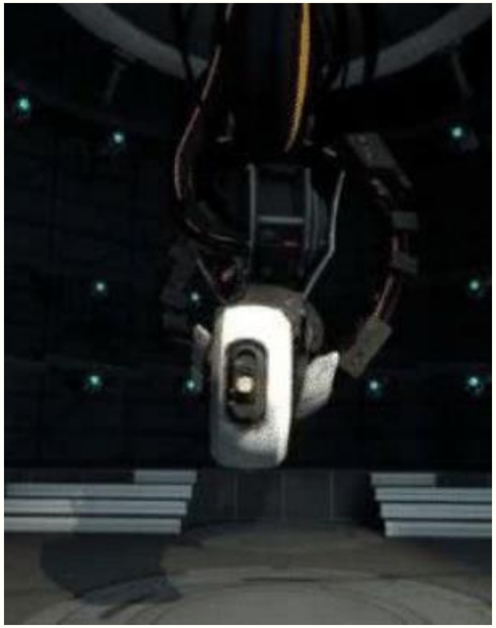
Machine Learning
Lecture1: Introduction

Jie Li

nijanice@163.com

Intelligent Machine in science fiction films


Self-Driving-Car

http://www.iqiyi.com/w_19rskt3y5l.html

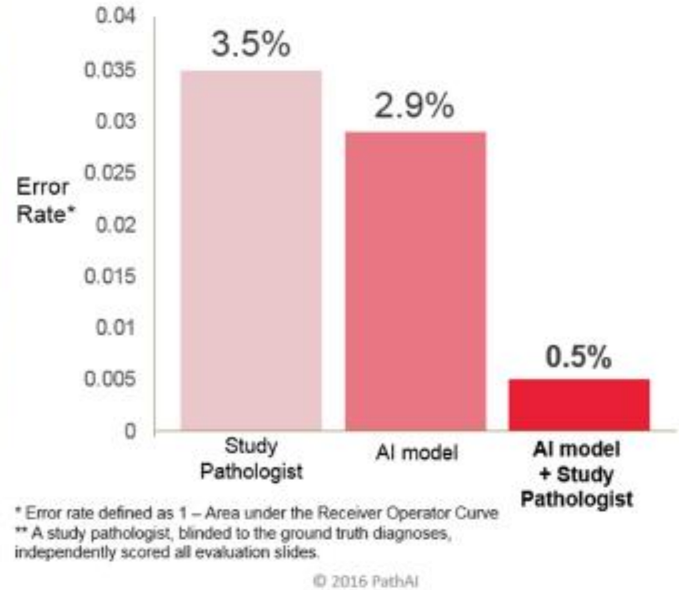
Robotics Control

- Ping pong robot
  - https://www.youtube.com/watch?v=tIIJME8-au8

Medical Image Analysis

- Breast Cancer Diagnoses

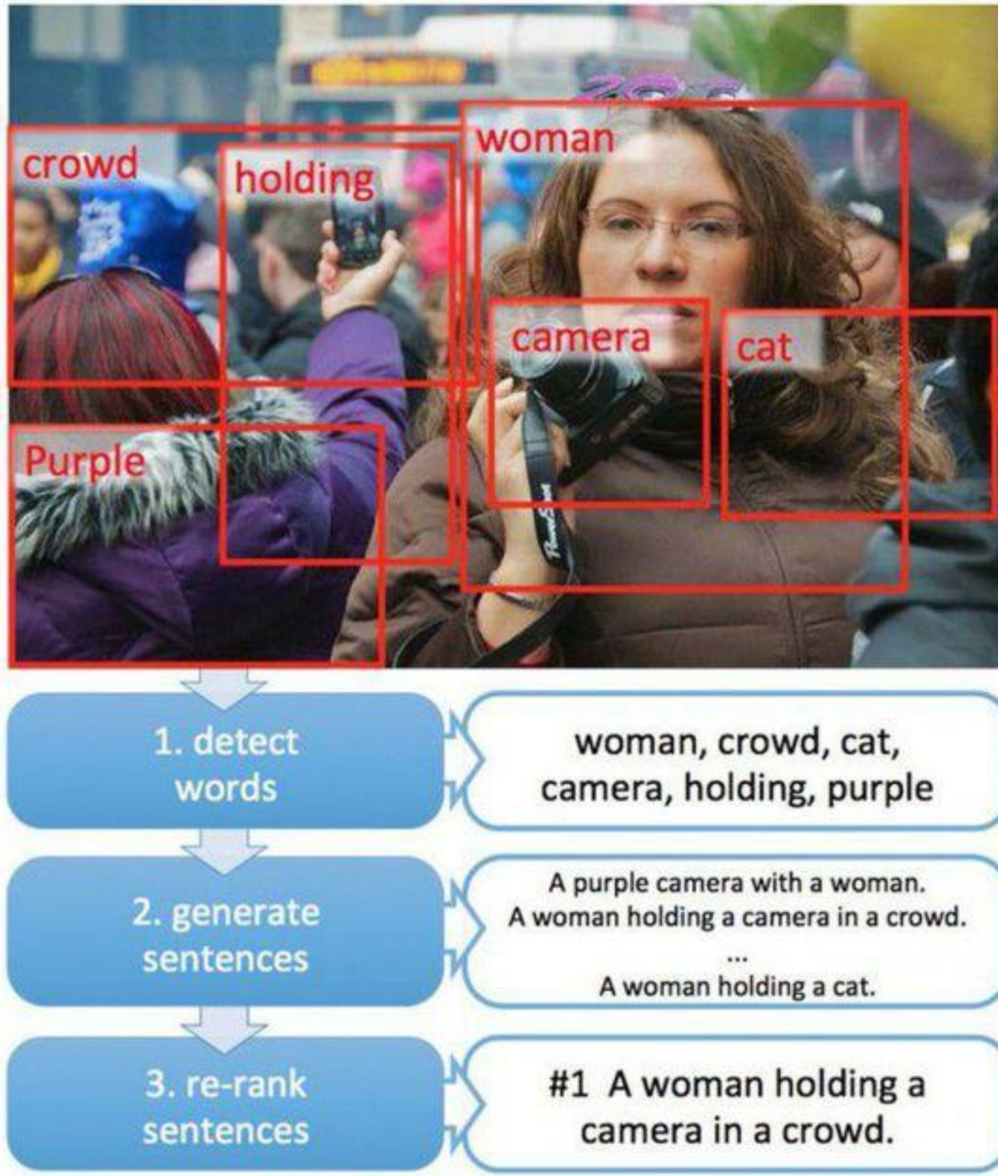
(AI + Pathologist) > Pathologist



Wang, Dayong, et al. "Deep learning for identifying metastatic breast cancer." arXiv preprint arXiv:1606.05718 (2016).
https://blogs.nvidia.com/blog/2016/09/19/deep-learning-breast-cancer-diagnosis/

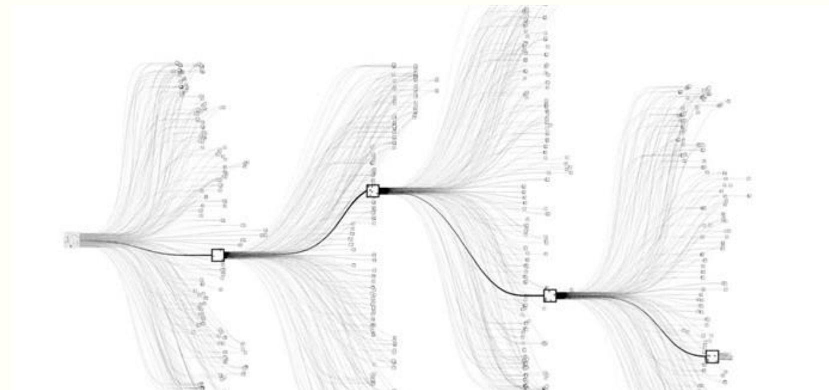
Microsoft Artificial Brain With 'Project Adam'

Spoken Natural Language Interfaces



AlphaGo


nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card forty years on
PAGE 462
NATURE.COM/NATURE




巴士单机游戏

News Recommendation

美国大选 + 关注

“特朗普时代”的中美新局

周浩：美国在经济上强势可能带来外交上相对弱势，美国可能给予中国在亚太更大的主动权，以为经济发展蓄势。

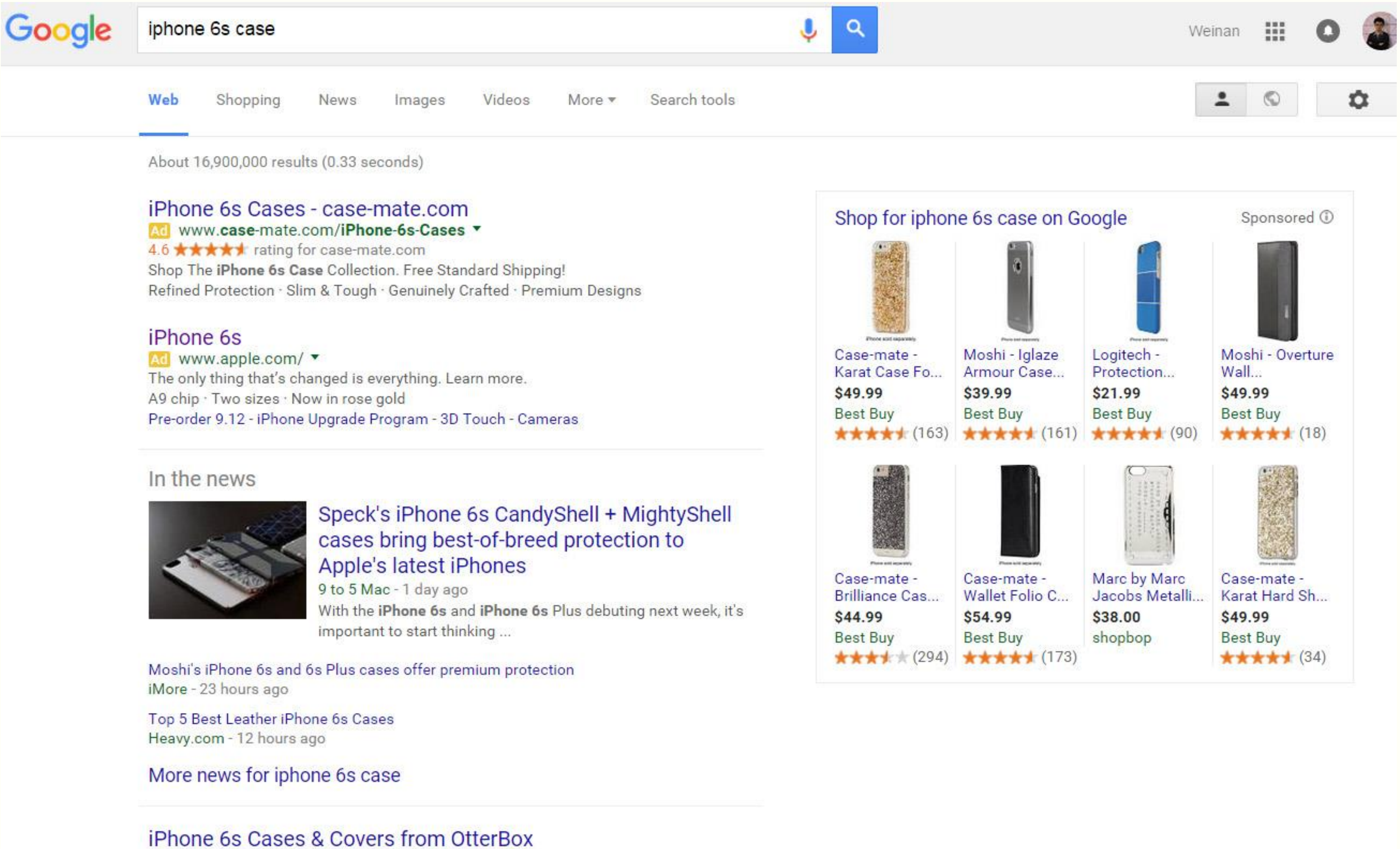
更新于2016年11月16日 07:07 德国商业银行首席中国经济师 周浩 为英国《金融时报》中文网撰稿

特朗普强势当选美国总统，给全市场留下了一个要解的难题：到底这位特立独行的美国白人会给世界带来怎样的变化，而未来世界格局中，中美两大经济体又将会以怎样的方式来进行互动。

到目前为止，我们只能通过特朗普在竞选过程中的讲话，部分了解未来美国政策的走向。比如说，特朗普反对TPP，认为目前的全球化策略并没有能够解决美国企业的困境，并表示要对中国商品征收45%的关税，同时要在美国和墨西哥边境建造"长城"来防止非法移民。特朗普也反对美国目前的世界警察角色，认为这给美国普通家庭带来了负担和悲痛，这意味着美国在全球战略布局中将更多采取收缩策略。此外，特朗普认为美国的能源政策和医疗保险制度是个灾难，认为政府插手太多，造成了巨大的浪费。

- Predict whether a user will like a news given its reading context

Online Advertising



Text Generation

- Making decision of selecting the next word/char
- Chinese poem example. Can you distinguish?

Human	Machine
南陌春风早，东邻去日斜。	山夜有雪寒，桂里逢客时。
紫陌追随日，青门相见时。	此时人且饮，酒愁一节梦。
胡风不开花，四气多作雪。	四面客归路，桂花开青竹。

Yu, Lantao, et al. "Seqgan: sequence generative adversarial nets with policy gradient." *AAAI* 2017.

Methodologies of AI

- Rule-based
  - Implemented by direct programing
  - Inspired by human heuristics

- Data-based
  - Expert systems
    - Experts or statisticians create rules of predicting or decision making based on the data
  - Machine learning
    - Direct making prediction or decisions based on the data
    - Data Science

Related Disciplines

- Artificial Intelligence
- Data Mining
- Probability and Statistics
- Information theory
- Numerical optimization
- Psychology (developmental, cognitive)
- Neurobiology
- Philosophy
- Computational complexity theory
- Control theory (adaptive)
- ….

What is Machine Learning?

- Learning is any process by which a system improves performance from experience

--- Herbert Simon

Turing Award (1975)

artificial intelligence, the psychology of human cognition

Nobel Prize in Economics (1978)

decision-making process within economic organizations

What is Machine Learning?

- A more mathematical definition by Tom Mitchell

- Machine learning is the study of algorithms that
  - improvement their performance *P*
  - at some task *T*
  - based on experience *E*
  - with non-explicit programming
- A well-defined learning task is given by <*P*, *T*, *E*>

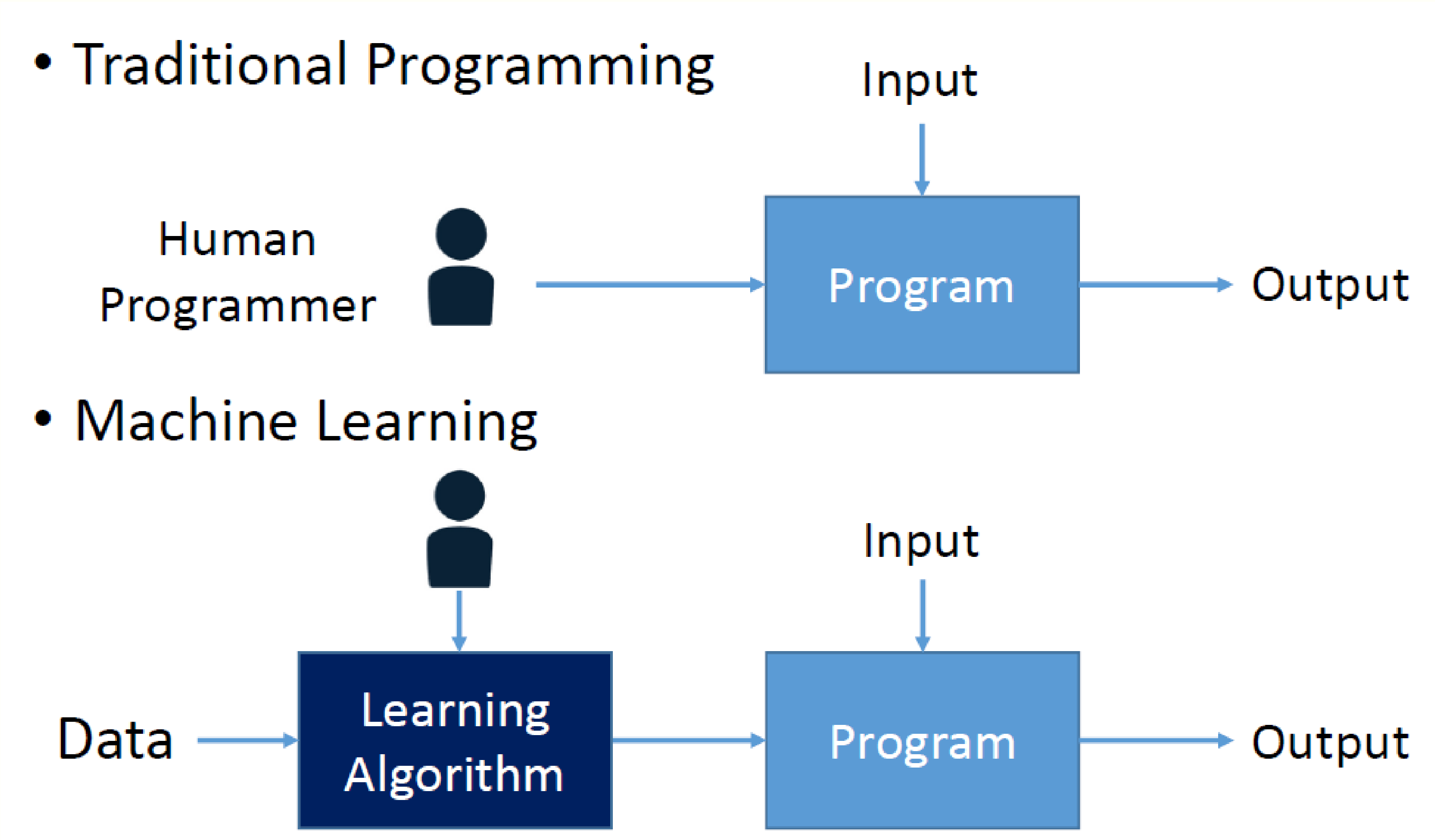
Learning is used when

- Develop systems that are too difficult/expensive to construct manually because they require specific detailed skills or knowledge tuned to a specific task （Speech / face recognition, game of Go）
- Even if we had a good idea about how to do it, the program might be horrendously complicated. (Robot arm)
- Human expertise does not exist (navigating on Mars)
- Solution changes in time (routing on a computer network)

Learning is used when

- Develop systems that can automatically adapt and customize themselves to individual users.
  - Personalized news or mail filter
  - Personalized tutoring
- Discover new knowledge from large databases (***data mining***).
  - Market basket analysis (e.g. diapers and beer)
  - Medical text mining (e.g. migraines to calcium channel blockers to magnesium)

Programming vs. Machine Learning



Why Study Machine Learning? The Time is Ripe

- Many basic effective and efficient algorithms available.
- Large amounts of on-line data available.
- Large amounts of computational resources available.

What is Learning?

- Herbert Simon: “Learning is any process by which a system improves performance from experience.”
- What is the task?
  - Classification
  - Problem solving / planning / control

Classification

- Assign object/event to one of a given finite set of categories.
  - ?

Classification

- Assign object/event to one of a given finite set of categories.
  - Medical diagnosis
  - Credit card applications or transactions
  - Fraud detection in e-commerce
  - Worm detection in network packets
  - Spam filtering in email
  - Recommended articles in a newspaper
  - Recommended books, movies, music, or jokes
  - Financial investments
  - DNA sequences
  - Spoken words
  - Handwritten letters
  - Astronomical images

Problem Solving / Planning / Control

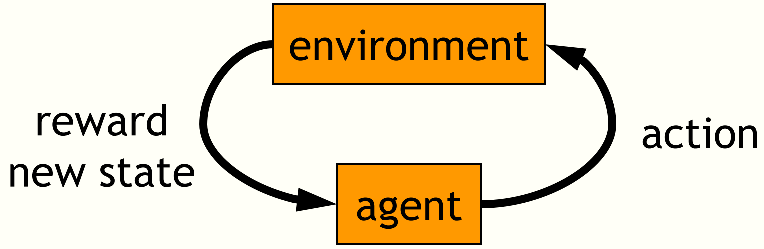
- Performing actions in an environment in order to achieve a goal.
  - ?

Problem Solving / Planning / Control

- Performing actions in an environment in order to achieve a goal.
  - Playing checkers, chess
  - Driving a car or a jeep
  - Flying a plane, helicopter, or rocket
  - Controlling an elevator
  - Controlling a character in a video game
  - Controlling a mobile robot

Types of learning task

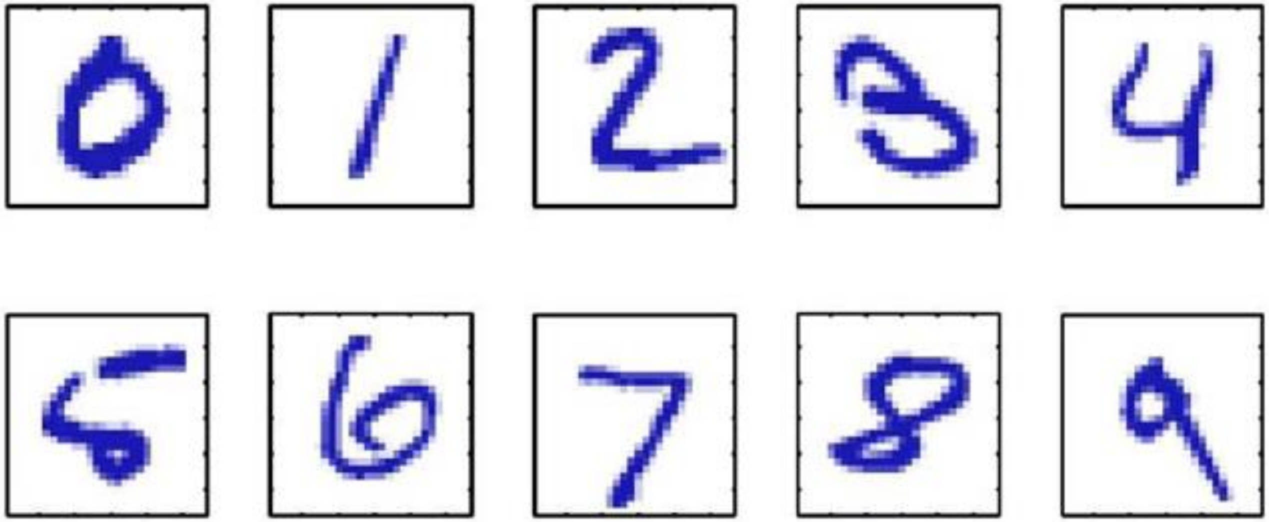
- Supervised learning
  - infer a function from labeled training data.
- Unsupervised learning
  - try to find hidden structure in unlabeled training data
  - clustering
- Reinforcement learning
  - To learn a policy of taking actions in a dynamic environment and acquire rewards



Machine Learning Problems

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Example 1: hand-written digit recognition



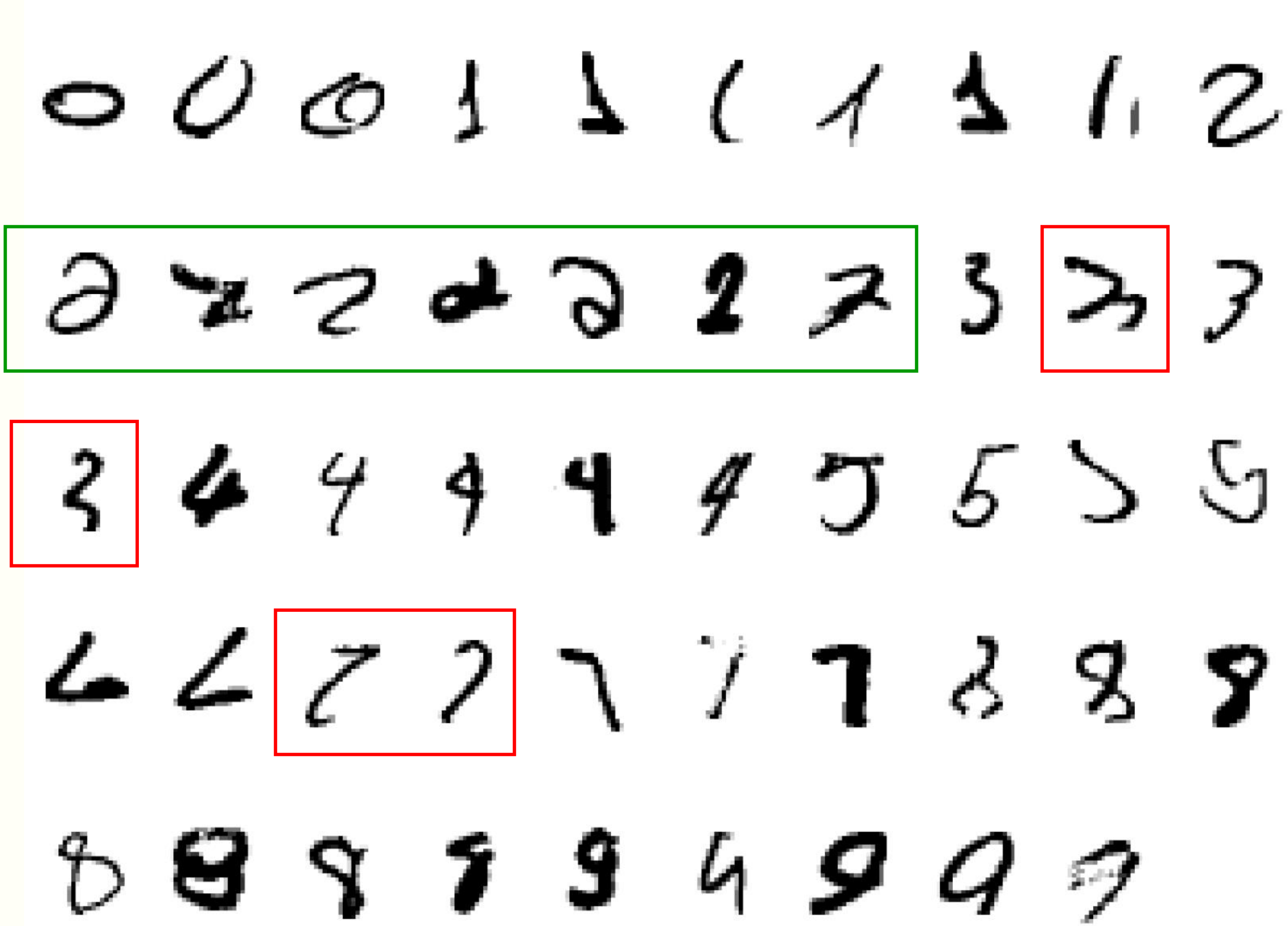
Images are 28 x 28 pixels

Represent input image as a vector $\mathbf{x} \in \mathbb{R}^{784}$

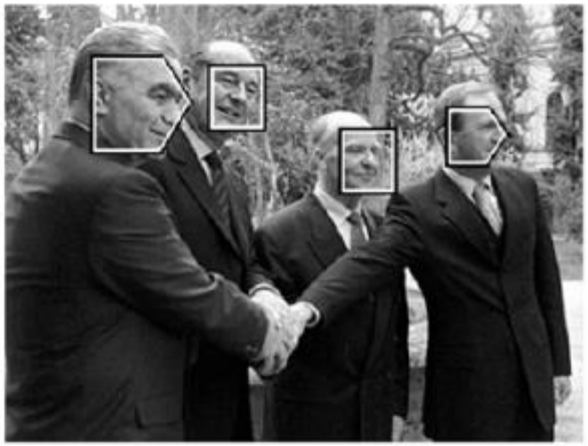
Learn a classifier $f(\mathbf{x})$ such that,

$$f : \mathbf{x} \to \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

A classic example of a task that requires machine learning: It is very hard to say what makes a 2



Example 2: Face detection



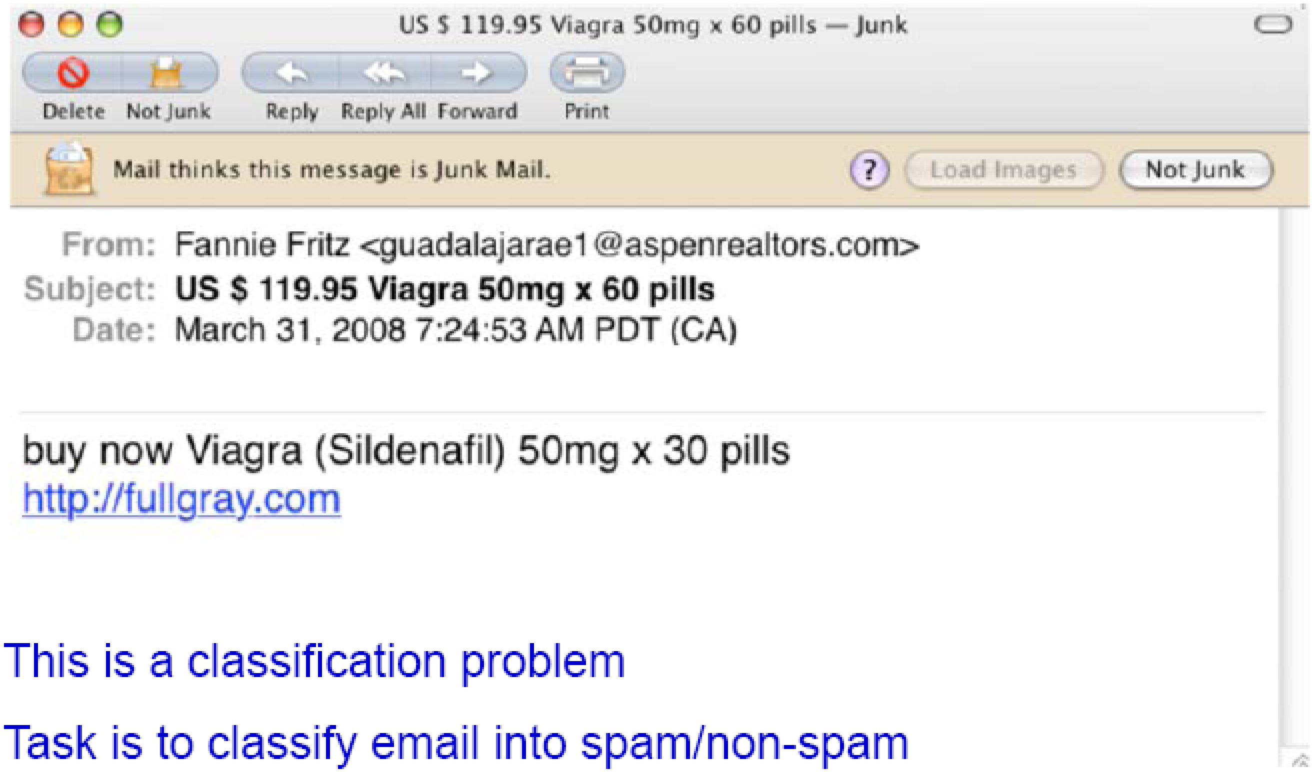
- Again, a supervised classification problem
- Need to classify an image window into three classes:
  - non-face
  - frontal-face
  - profile-face

Classifier is learnt from labelled data

Training data for frontal faces

- 5000 faces
  - All near frontal
  - Age, race, gender, lighting
- 10^8 non faces
- faces are normalized
  - scale, translation

Example 3: Spam detection



- This is a classification problem
- Task is to classify email into spam/non-spam
- Data x_i is word count, e.g. of viagra, outperform, “you may be surprized to be contacted” …
- Requires a learning system as “enemy” keeps innovating

Regression Applications

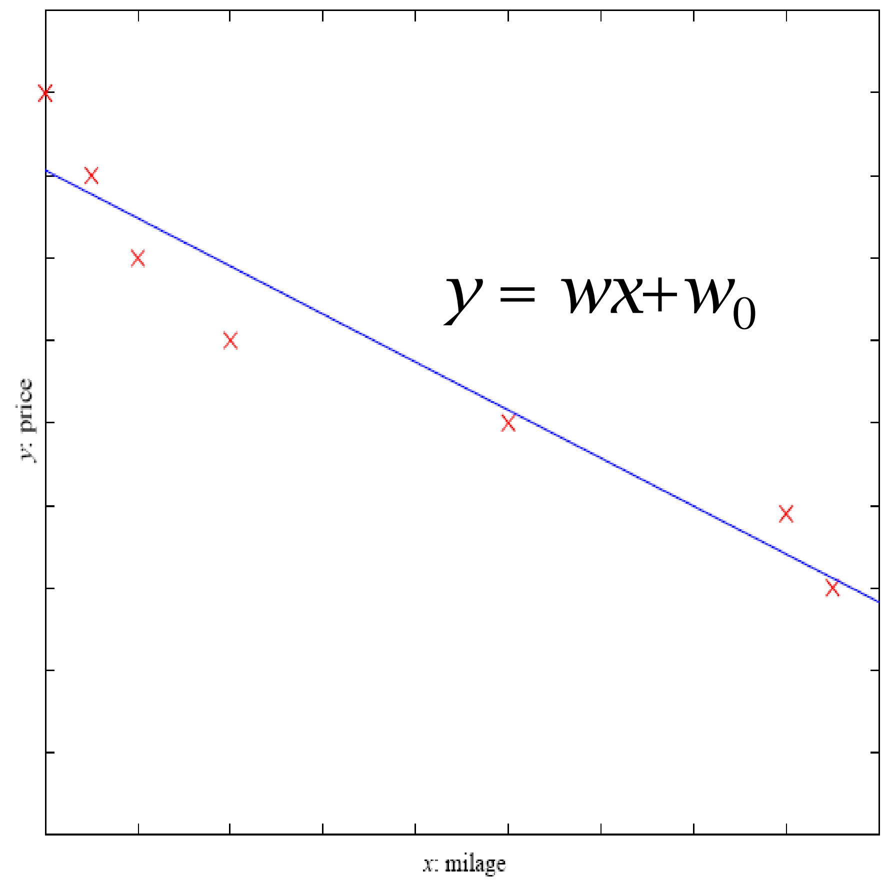
- Example: Price of a used car
- x : car attributes

  y : price

  $$y = g(x \mid \theta)$$

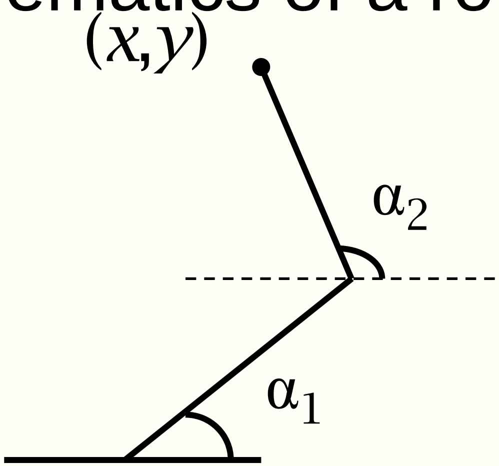
  $g(\ )$ model,

  θ parameters



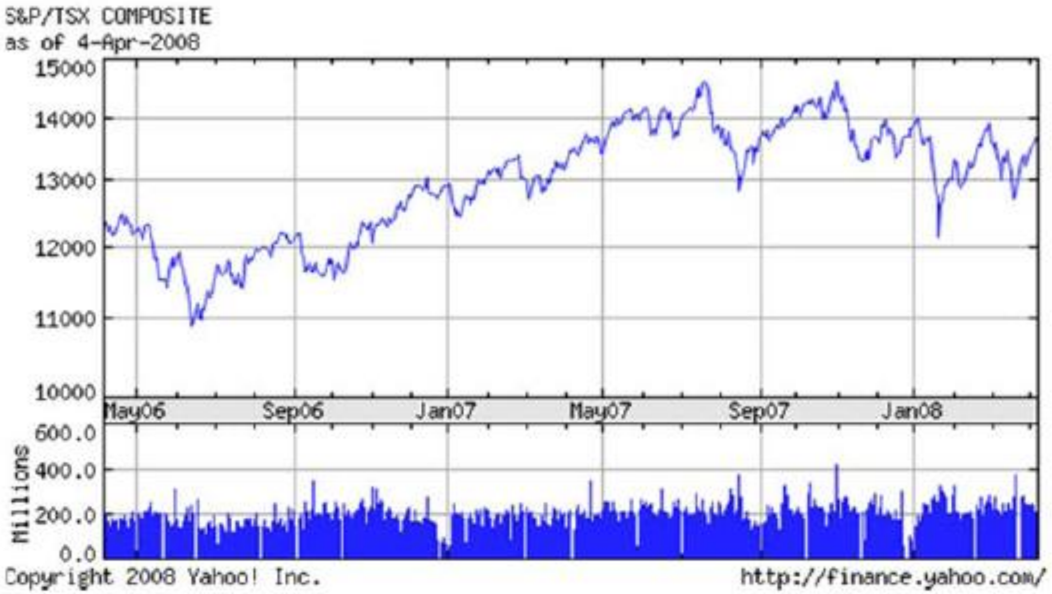
Regression Applications

- Navigating a car: Angle of the steering wheel
- Kinematics of a robot arm



$$\alpha_1 = g_1(x,y)$$
$$\alpha_2 = g_2(x,y)$$

Example 4: Stock price prediction

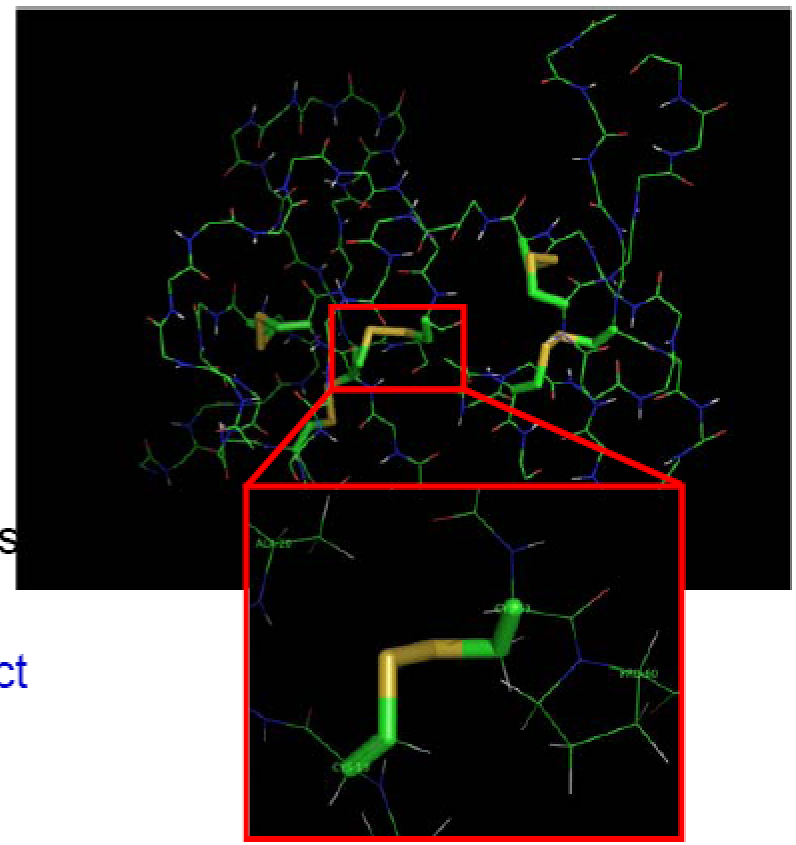


- Task is to predict stock price at future date
- This is a regression task, as the output is continuous

Example 5: Computational biology

$\mathbf{x}$ $\mathbf{y}$

AVITGACERDLQCG
KGTCCAVSLWIKSV
RVCTPVGTSGEDCH
PASHKIPFSGQRMH
HTCPCAPNLACVQT
SPKKFKCLSK



Protein Structure and Disulfide Bridges

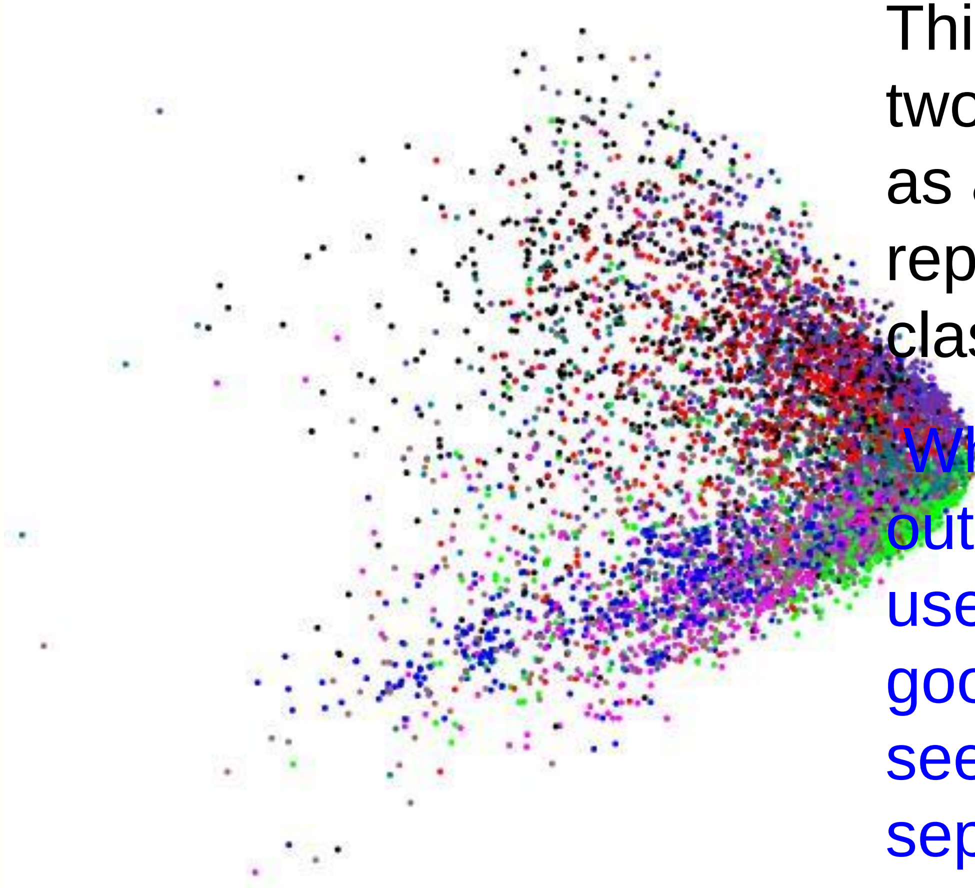
Regression task: given sequence predict 3D structure

Protein: 1IMT

Web examples: Recommender systems

People who bought Hastie …

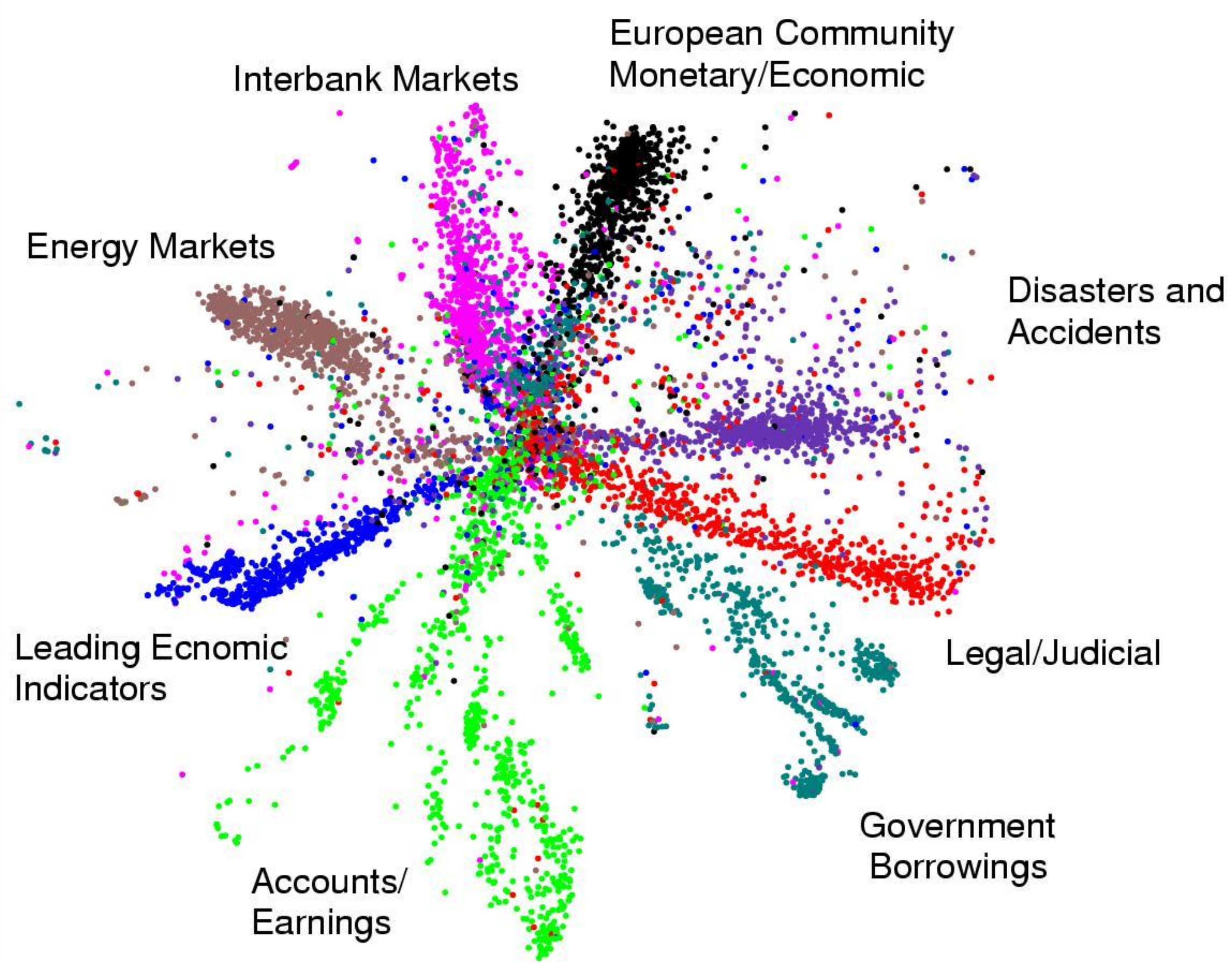
Displaying the structure of a set of documents using Latent Semantic Analysis (a form of PCA)



Each document is converted to a vector of word counts. This vector is then mapped to two coordinates and displayed as a colored dot. The colors represent the hand-labeled classes.

When the documents are laid out in 2-D, the classes are not used. So we can judge how good the algorithm is by seeing if the classes are separated.

Displaying the structure of a set of documents



History of Machine Learning

- 1950s
  - Samuel’s checker player
  - Selfridge’s Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston’s arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan’s ID3
  - Mathematical discovery with AM

History of Machine Learning

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology
- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning
  - Support vector machines
  - Kernel methods

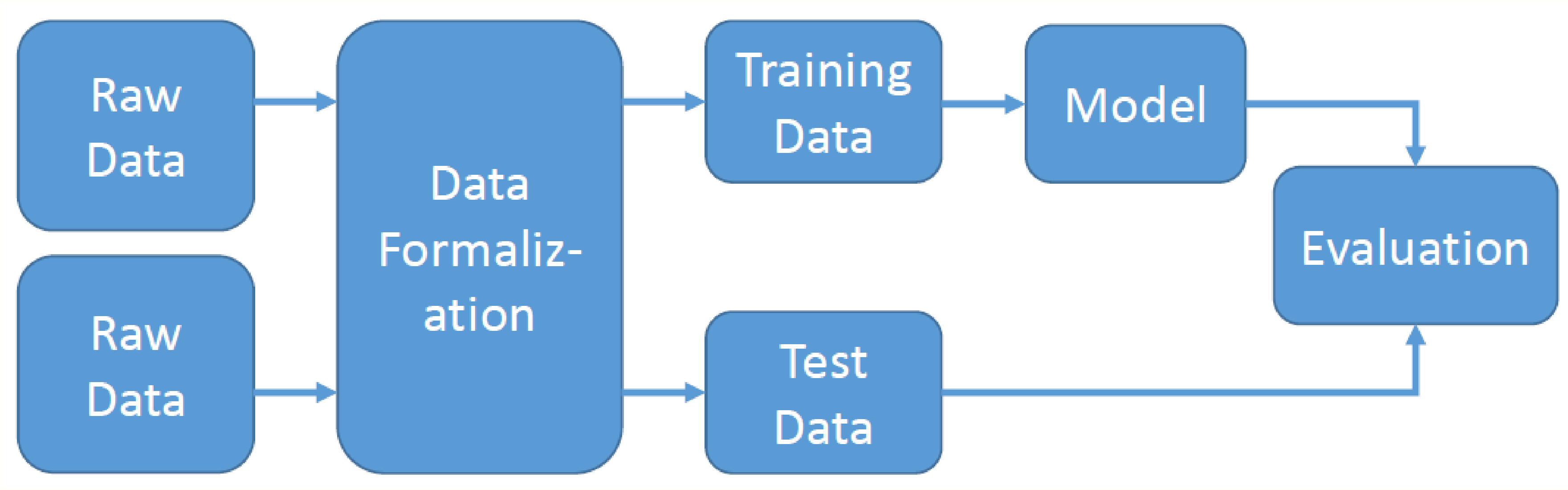
History of Machine Learning

- 2000s
  - Graphical models
  - Variational inference
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications
  - Compilers
  - Debugging
  - Graphics
  - Security (intrusion, virus, and worm detection)
  - Email management
  - Personalized assistants that learn
  - Learning in robotics and vision

History of Machine Learning

- 2010s
  - Deep learning
  - Learning from big data
  - Learning with GPUs or HPC
  - Multi-task & lifelong learning
  - Deep reinforcement learning
  - Massive applications to vision, speech, text, networks, behavior etc.
  - …

Machine Learning Process



- Basic assumption: there exist the same patterns
- across training and test data

Slide credit: L. Lazebnik

Supervised Learning

- Given the training dataset of (data, label) pairs,

$$D = \{(x_i, y_i)\}_{i=1,2,\dots,N}$$

  let the machine learn a function from data to label

$$y_i \simeq f_\theta(x_i)$$

- Function set $\{f_\theta(\cdot)\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ
- How to learn?
  - Update the parameter to make the prediction closed to the corresponding label
    - What is the learning objective?
    - How to update the parameters?

Learning Objective

- Make the prediction closed to the corresponding label

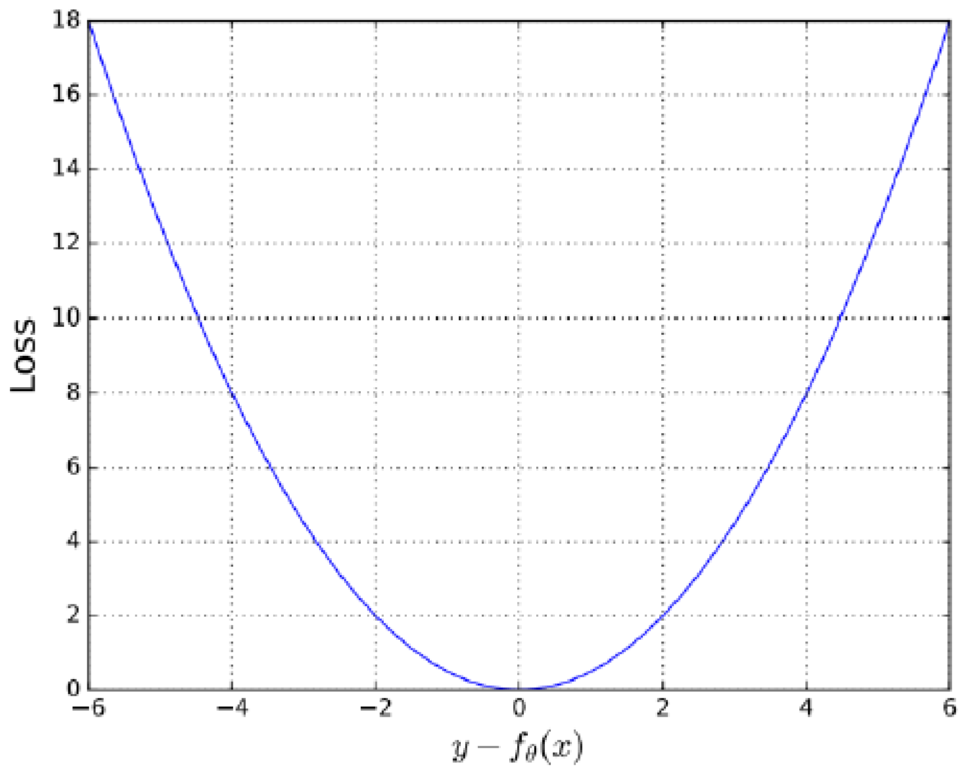
$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_\theta(x_i))$$

- Loss function $\mathcal{L}(y_i, f_\theta(x_i))$ measures the error between the label and prediction
- The definition of loss function depends on the data and task
- Most popular loss function: squared loss

$$\mathcal{L}(y_i, f_\theta(x_i)) = \frac{1}{2}(y_i - f_\theta(x_i))^2$$

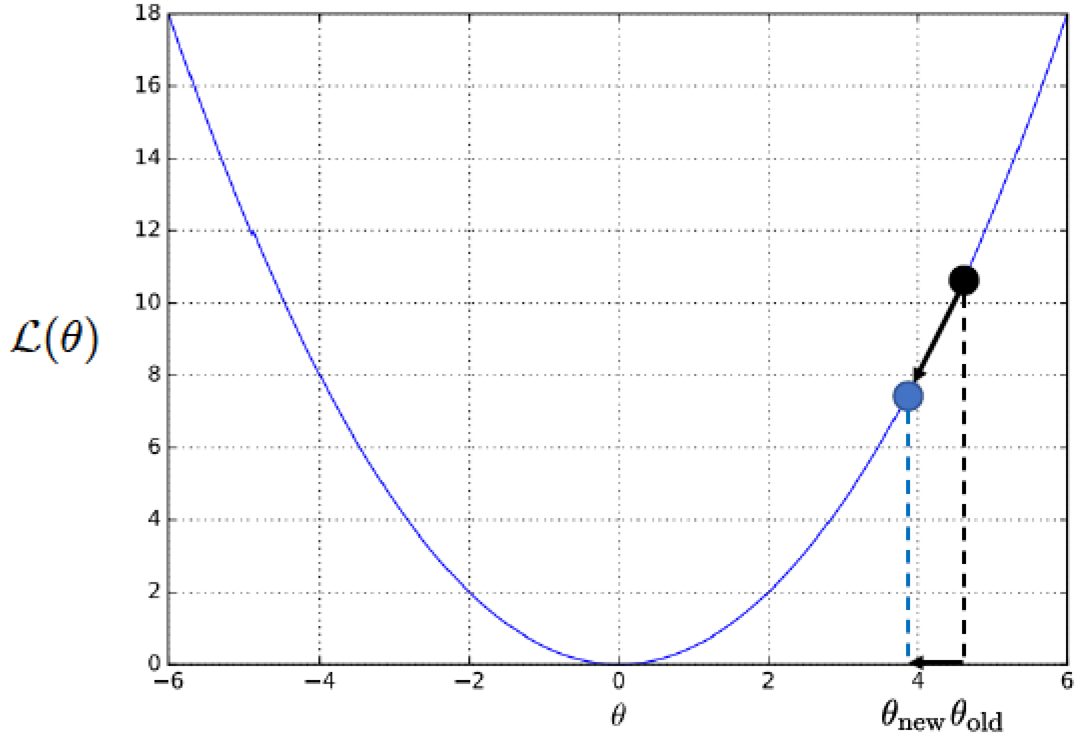
Squared Loss

$$\mathcal{L}(y_i, f_\theta(x_i)) = \frac{1}{2}(y_i - f_\theta(x_i))^2$$



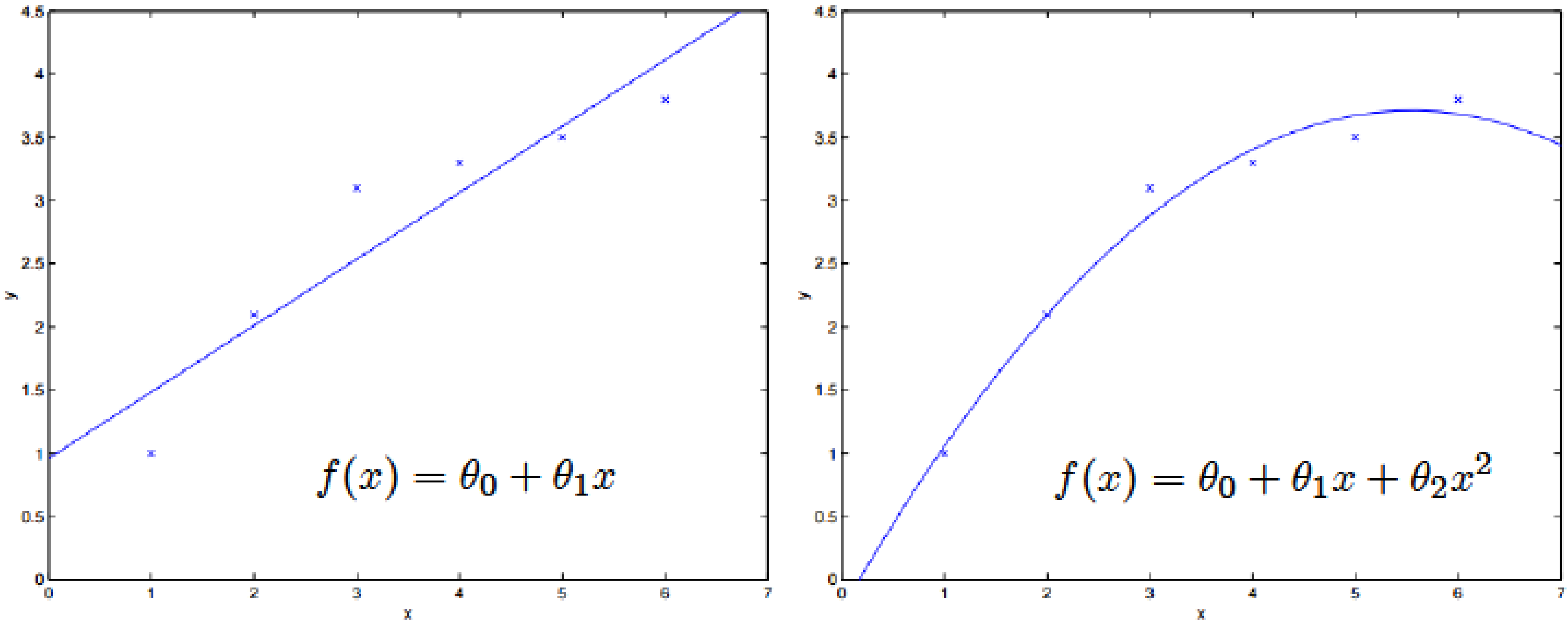
- Penalty much more on larger distances
- Accept small distance (error)
  - Observation noise etc.
  - Generalization

Gradient Learning Methods



$$\theta_{\text{new}} \leftarrow \theta_{\text{old}} - \eta \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

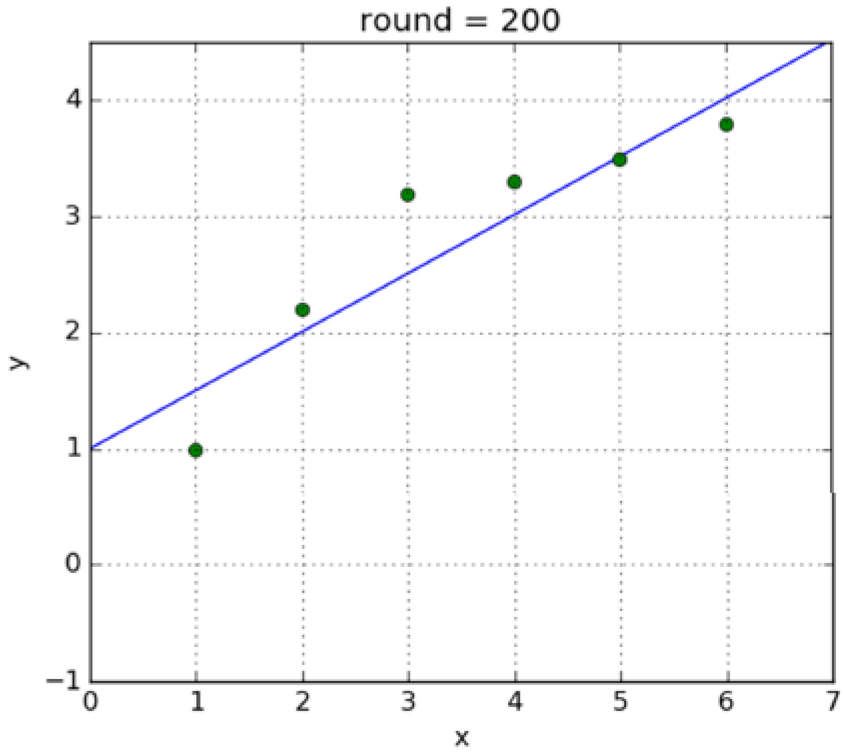
A Simple Example



- Observing the data $\{(x_i, y_i)\}_{i=1,2,\dots,N}$, we can use different models (hypothesis spaces) to learn
  - First, model selection (linear or quadratic)
  - Then, learn the parameters

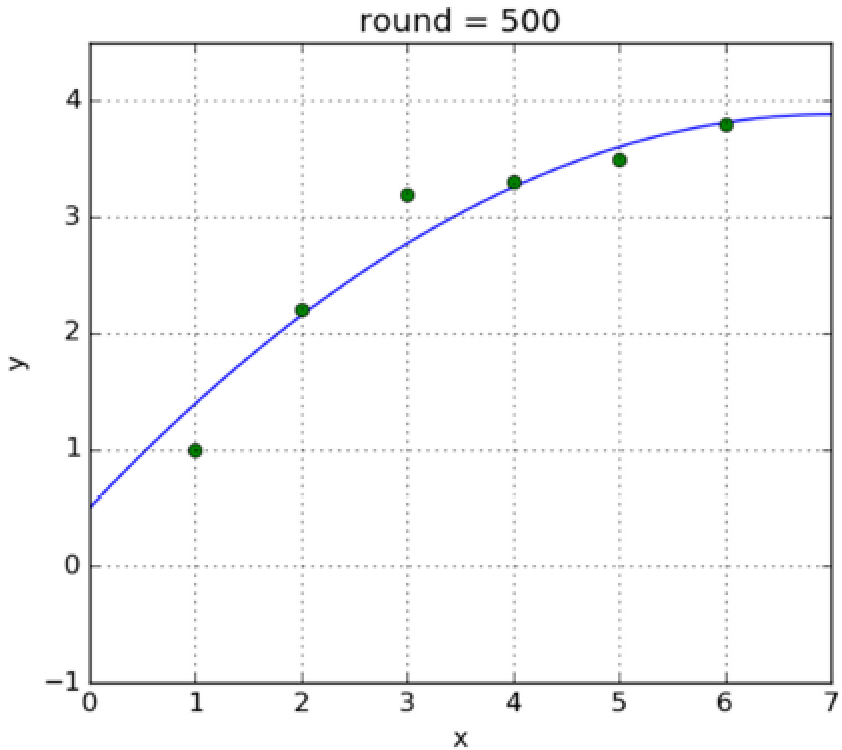
An example from Andrew Ng

Learning Linear Model - Curve



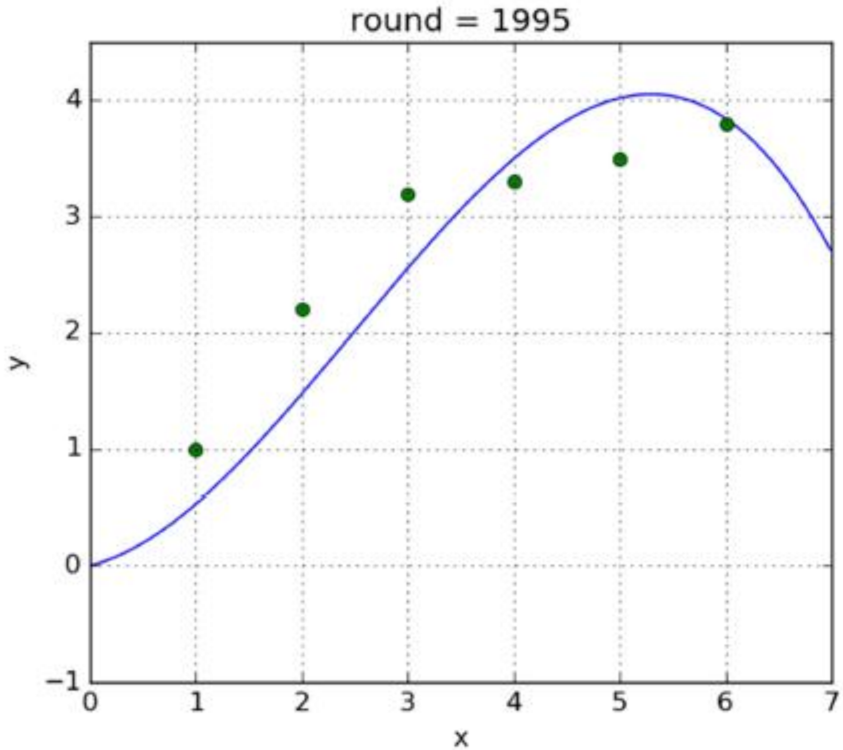
$$f(x) = \theta_0 + \theta_1 x$$

Learning Quadratic Model



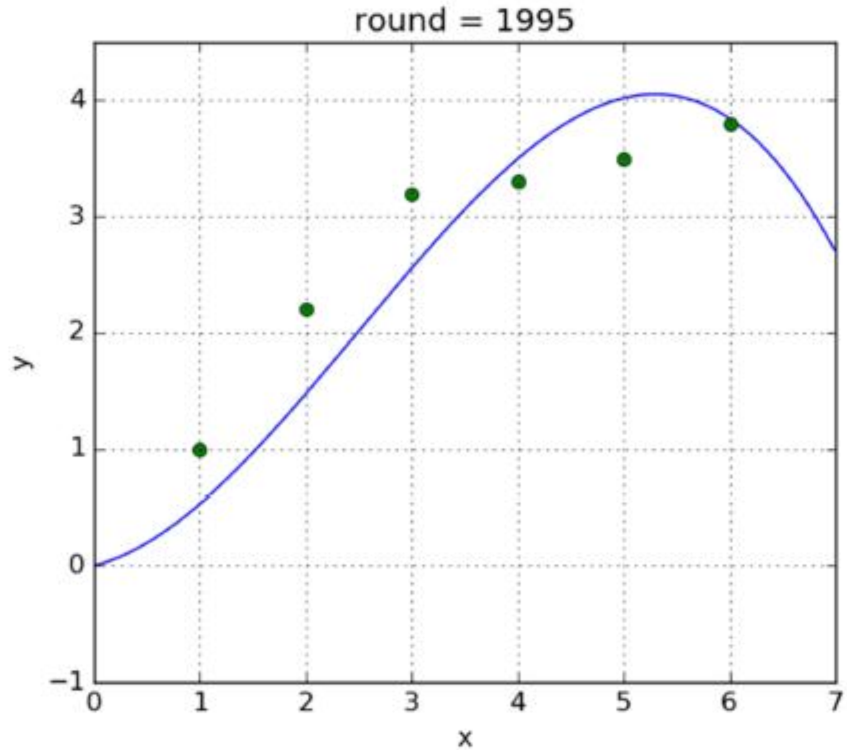
$$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

Learning Cubic Model



$$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

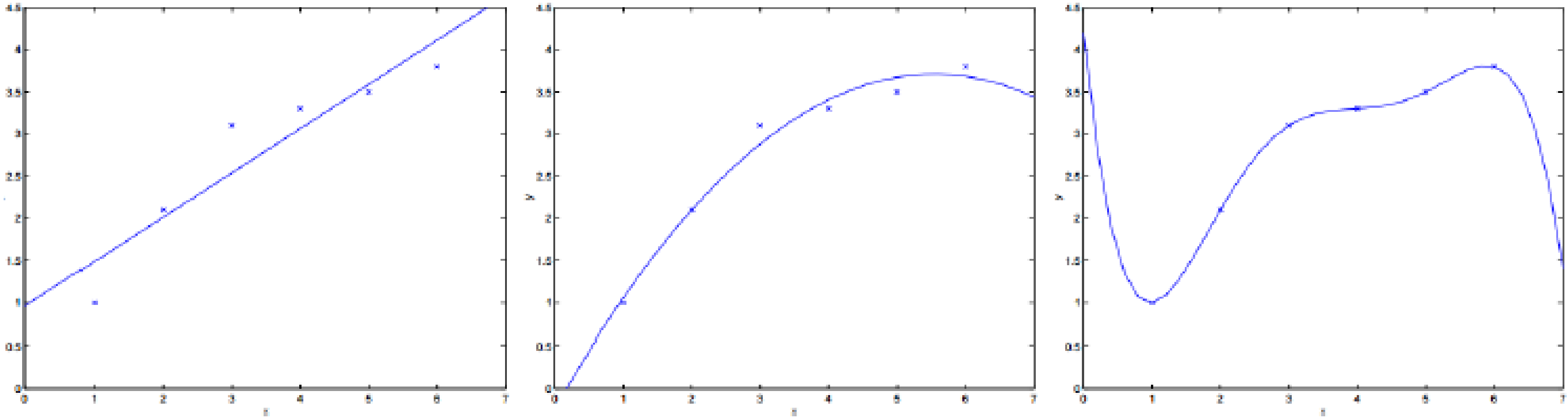
Learning Cubic Model



$$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

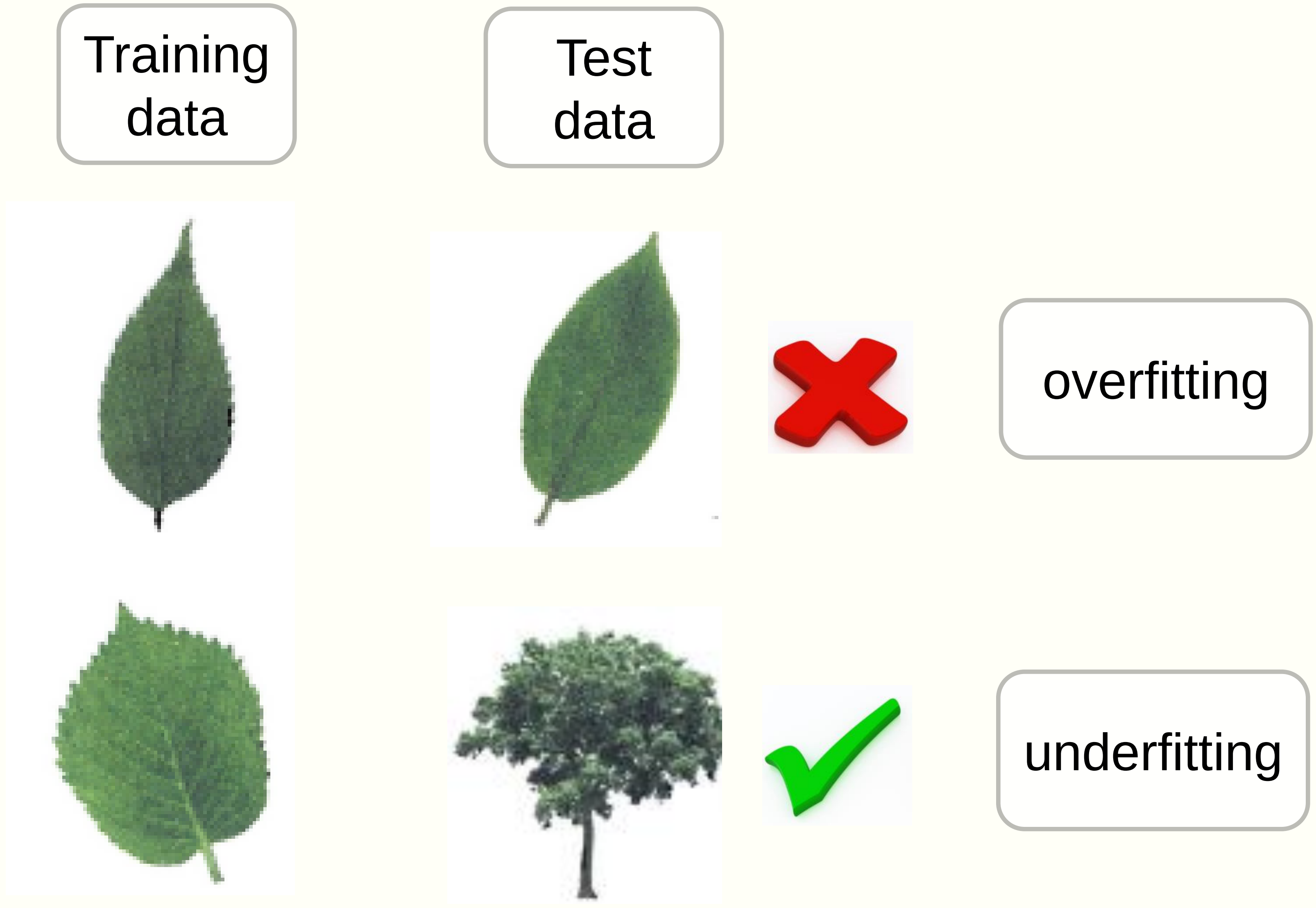
Model Selection

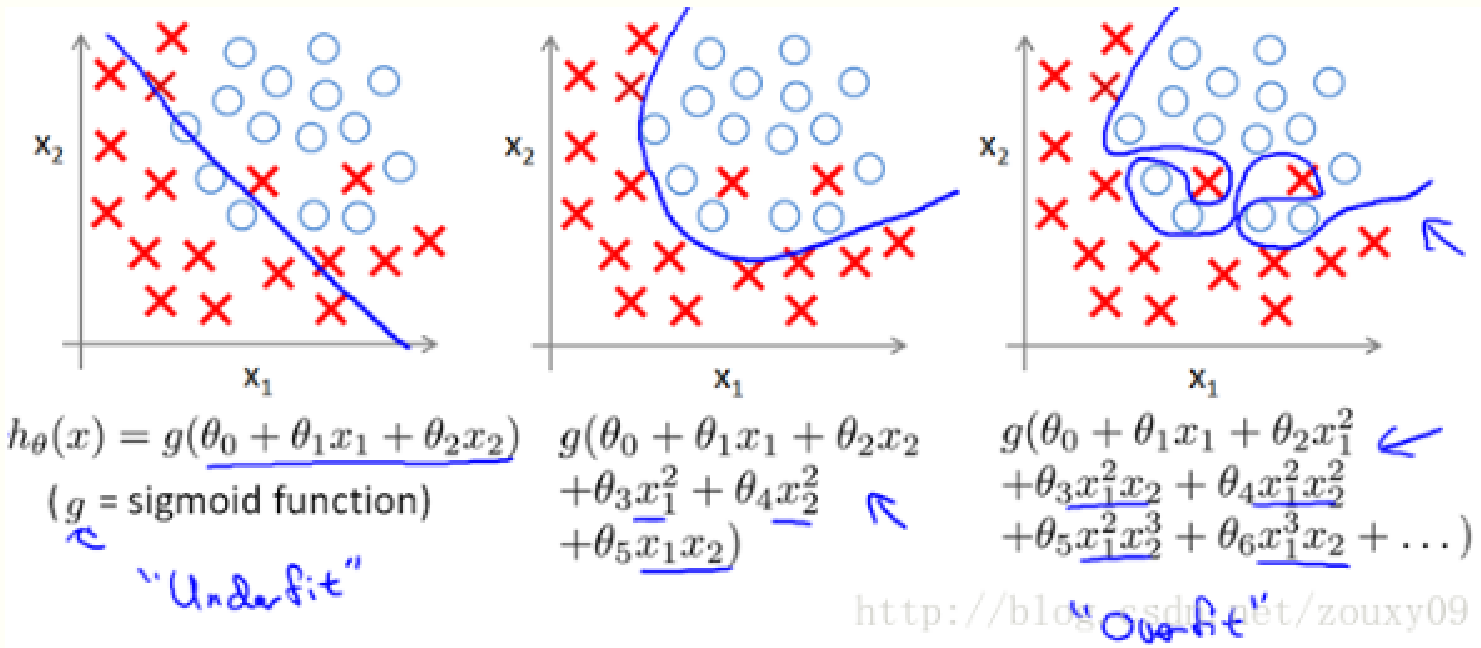
- Which model is the best?



Linear model: underfitting   Quadratic model: well fitting   5th-order model: overfitting

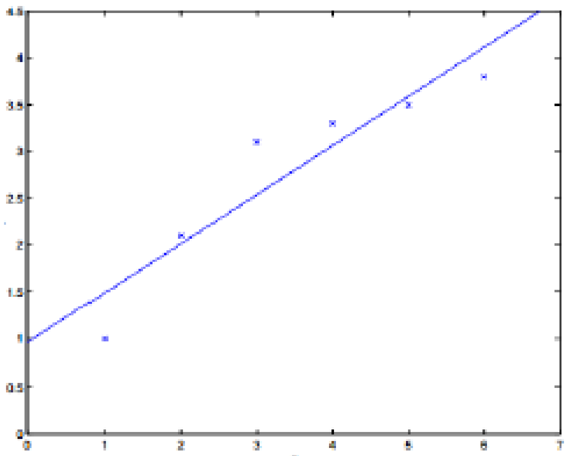
- Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data.
- Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship


Training data
Test data
overfitting
underfitting

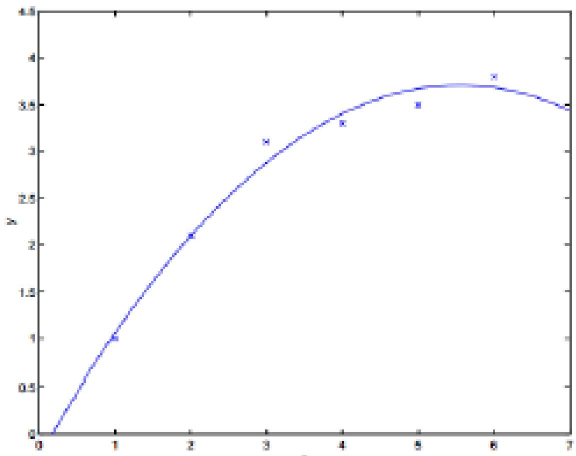

x_2
x_1
x_2
x_1
x_2
x_1
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
(g = sigmoid function)
"Underfit"
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$
$+\theta_3 x_1^2 + \theta_4 x_2^2$
$+\theta_5 x_1 x_2)$
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2$
$+\theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2$
$+\theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$
"Overfit"
http://blog.csdn.net/zouxy09

Model Selection

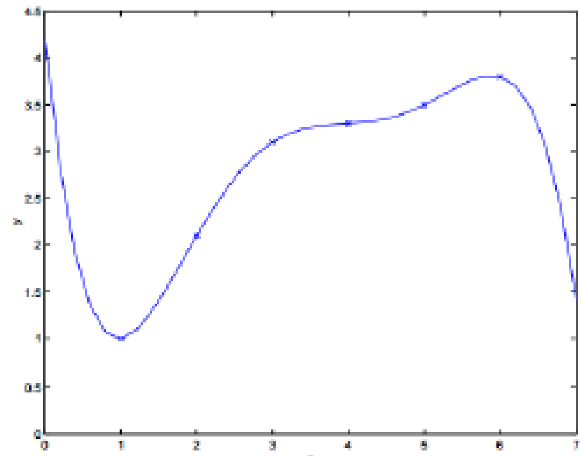
- Which model is the best?



Linear model: underfitting



Quadratic model: well fitting



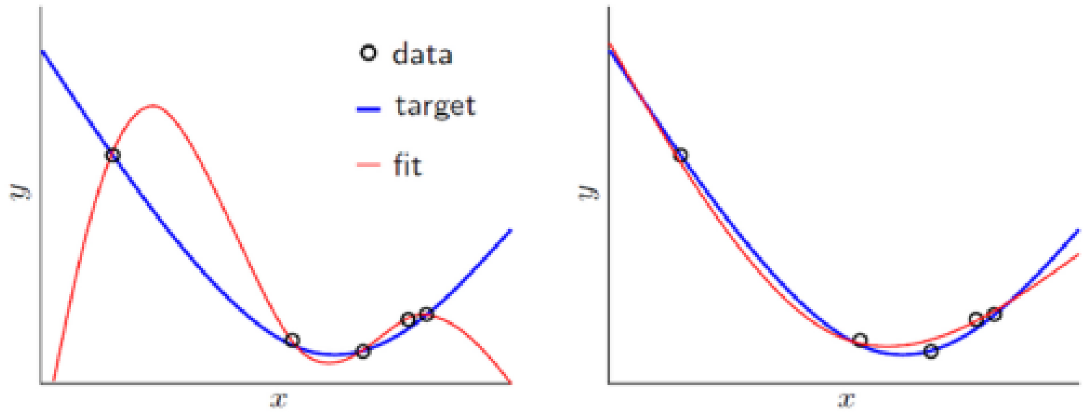
5^{th}-order model: overfitting

- Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data.
- Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship

Regularization

- Add a penalty term of the parameters to prevent the model from overfitting the data

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_{\theta}(x_i)) + \lambda \Omega(\theta)$$



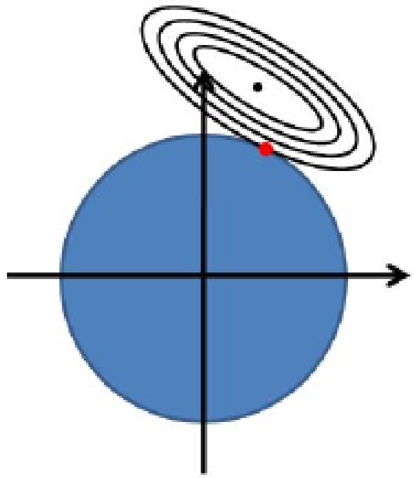
(a) without regularization (b) with regularization

Typical Regularization

- L2-Norm (Ridge)

$$\Omega(\theta) = ||\theta||_2^2 = \sum_{m=1}^{M} \theta_m^2$$

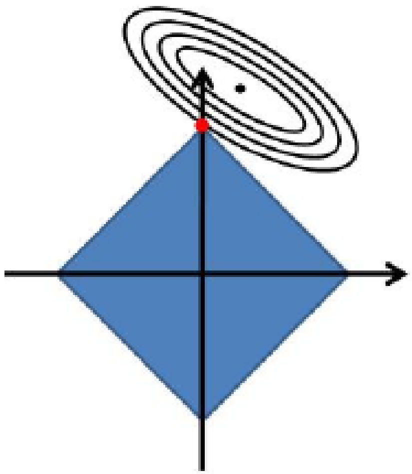
$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_{\theta}(x_i)) + \lambda ||\theta||_2^2$$



- L1-Norm (LASSO)

$$\Omega(\theta) = ||\theta||_1 = \sum_{m=1}^{M} |\theta_m|$$

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_{\theta}(x_i)) + \lambda ||\theta||_1$$



Principle of Occam's razor

Among competing hypotheses, the one with the fewest assumptions should be selected.

- Recall the function set $\{f_\theta(\cdot)\}$ is called hypothesis space

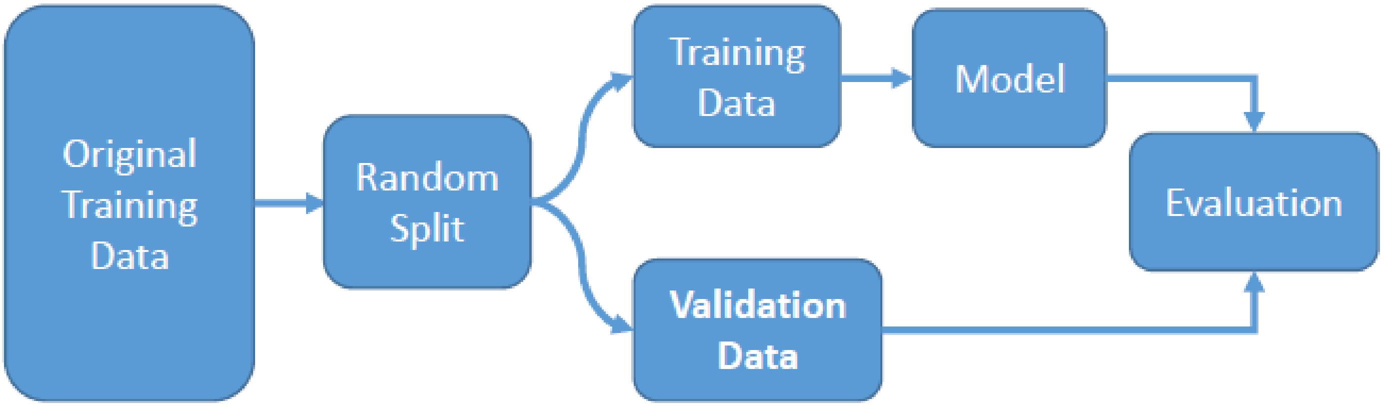
$$\min_\theta \underbrace{\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(y_i, f_\theta(x_i))}_{\text{Original loss}} + \lambda\underbrace{\Omega(\theta)}_{\text{Penalty on assumptions}}$$

Model Selection

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_\theta(x_i)) + \lambda ||\theta||_2^2$$

- An ML solution has model parameters θ and optimization hyperparameters λ
- Hyperparameters
  - Define higher level concepts about the model such as complexity, or capacity to learn.
  - **Cannot be learned directly from the data** in the standard model training process and need to be predefined.
  - Can be decided by setting different values, training different models, and choosing the values that test better
- Model selection (or hyperparameter optimization) cares how to select the optimal hyperparameters.

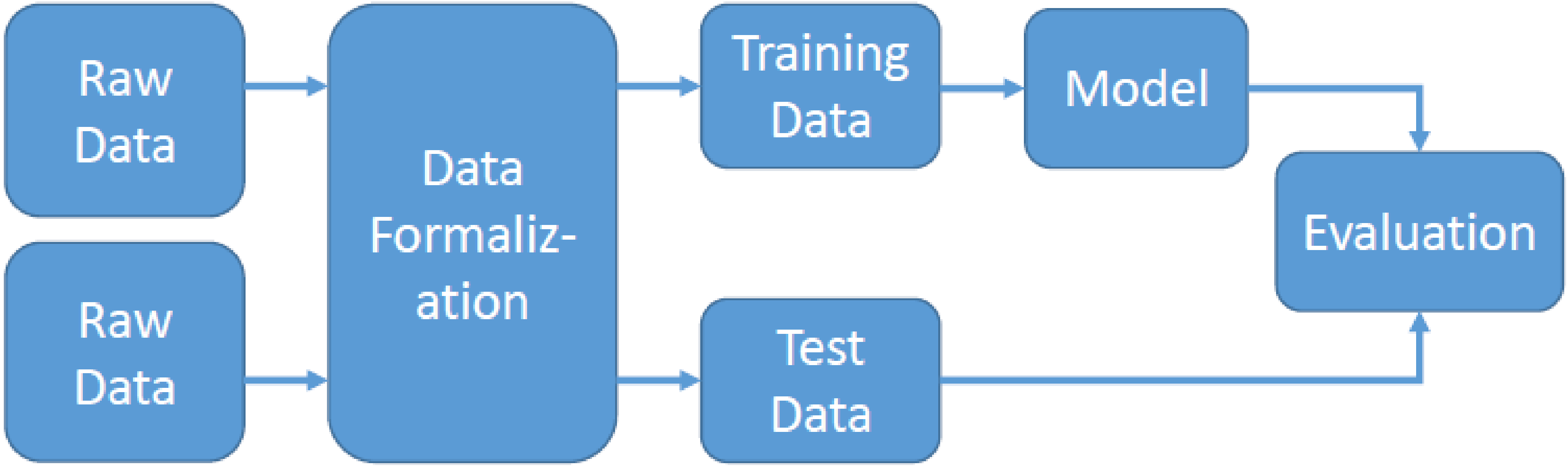
Cross Validation for Model Selection



K-fold Cross Validation

1. Set hyperparameters
2. For *K* times repeat:
   - Randomly split the original training data into training and validation datasets
   - Train the model on training data and evaluate it on validation data, leading to an evaluation score
3. Average the *K* evaluation scores as the model performance

Machine Learning Process



- After selecting 'good' hyperparameters, we train the model over the whole training data and the model can be used on test data.

Generalization Ability

- Generalization Ability is the model prediction capacity on unobserved data
  - Can be evaluated by Generalization Error, defined by

$$R(f) = \mathbb{E}[\mathcal{L}(Y, f(X))] = \int_{X \times Y} \mathcal{L}(y, f(x)) p(x, y) dx dy$$

  - where $p(x, y)$ is the underlying (probably unknown) joint data distribution

- Empirical estimation of GA on a training dataset is

$$\hat{R}(f) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f(x_i))$$

The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

The machine learning framework

$$y = f(\mathbf{x})$$

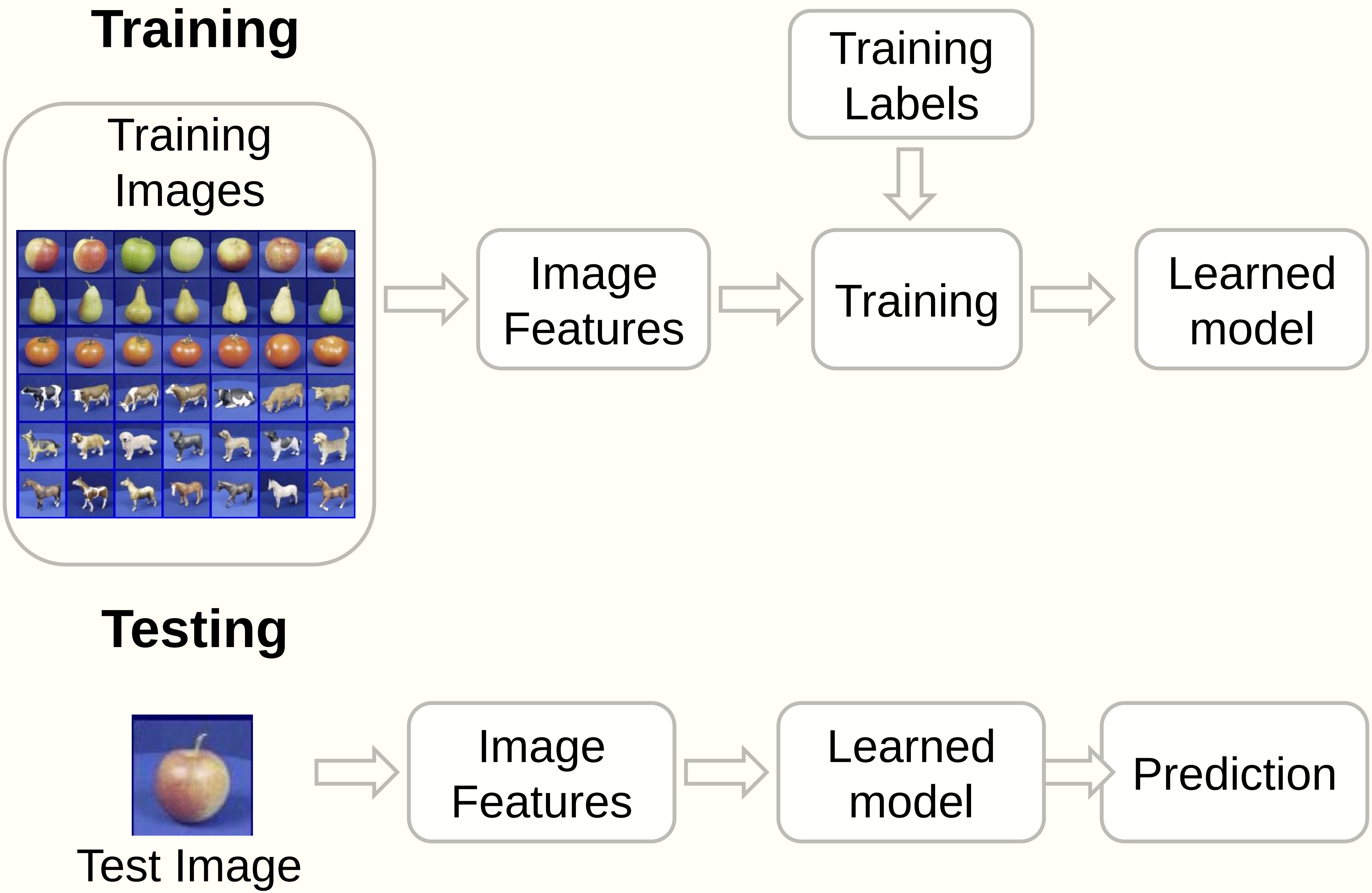
output

prediction function

Image feature

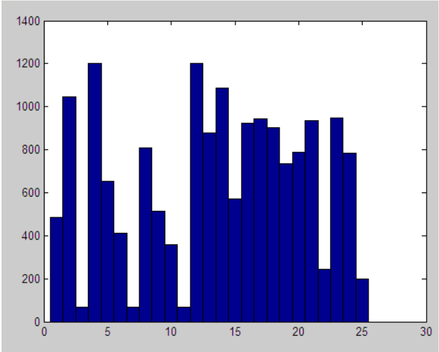
- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Steps

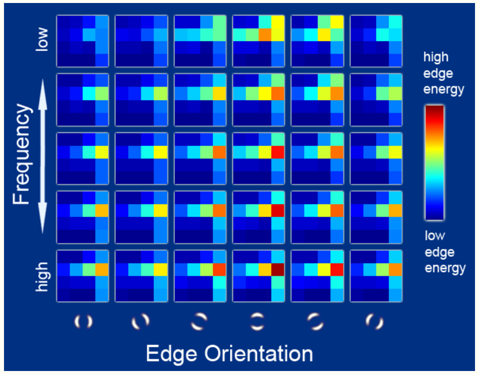

Training
Training Labels
Training Images
Image Features
Training
Learned model
Testing
Test Image
Image Features
Learned model
Prediction

Features

- Raw pixels



- Histogram



- GIST descriptors

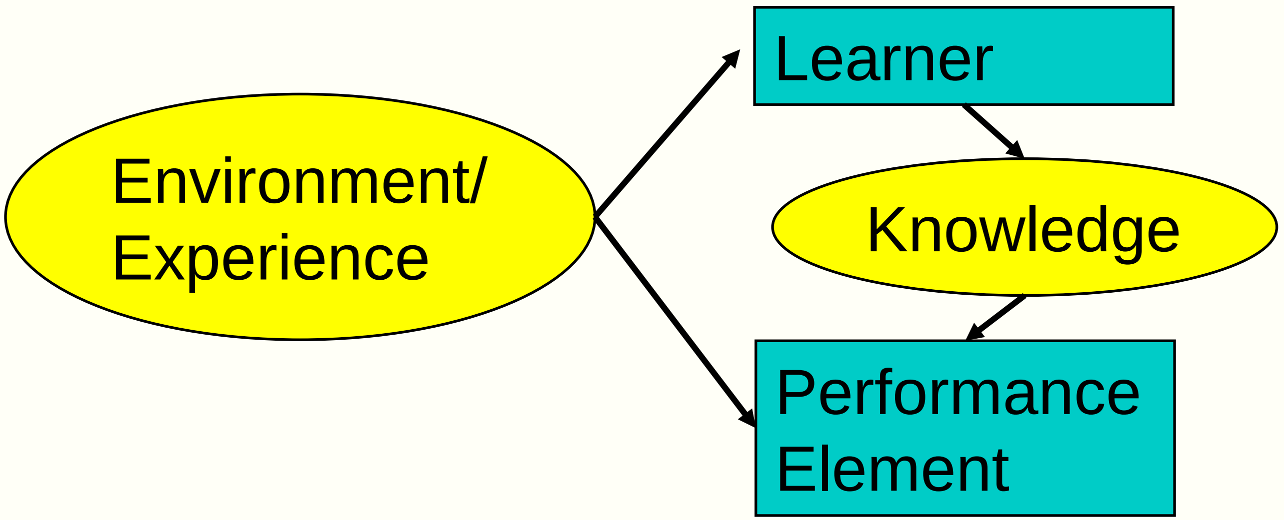
- …

Measuring Performance

- Classification Accuracy
- Solution correctness
- Solution quality (efficiency)
- Speed of performance

Designing a Learning System

- Choose the training experience
- Choose exactly what is to be learned, i.e. the ***target function***.
- Choose how to represent the target function.
- Choose a learning algorithm to infer the target function from the experience.



Defining the Learning Task

Improve on task, T, with respect to performance metric, P, based on experience, E.

T: Playing checkers
P: Percentage of games won against an arbitrary opponent
E: Playing practice games against itself

T: Recognizing hand-written words
P: Percentage of words correctly classified
E: Database of human-labeled images of handwritten words

T: Driving on four-lane highways using vision sensors
P: Average distance traveled before a human-judged error
E: A sequence of images and steering commands recorded while observing a human driver.

T: Categorize email messages as spam or legitimate.
P: Percentage of email messages correctly classified.
E: Database of emails, some with human-given labels

Sample Learning Problem

- Learn to play checkers from self-play
- We will develop an approach analogous to that used in the first machine learning system developed by Arthur Samuels at IBM in 1959.
- Rule(Video)

Training Experience

- Direct experience: Given sample input and output pairs for a useful target function.
  - Checker boards labeled with the correct move, e.g. extracted from record of expert play
- Indirect experience: Given feedback which is ***not*** direct I/O pairs for a useful target function.
  - Potentially arbitrary sequences of game moves and their final game results.
- Credit/Blame Assignment Problem: How to assign credit blame to individual moves given only indirect feedback?

Source of Training Data

- Good training examples selected by a "benevolent teacher."
- Provided random examples outside of the learner's control.
  - Learner can construct an arbitrary example and query an oracle for its label.
  - Learner can design and run experiments directly in the environment without any human guidance.
- …

Training vs. Test Distribution

- Generally assume that the training and test examples are independently drawn from the same overall distribution of data.
  - IID: Independently and identically distributed

Choosing a Target Function

- What function is to be learned and how will it be used by the performance system?
- For checkers, assume we are given a function for generating the legal moves for a given board position and want to decide the best move.
  - Could learn a function:

    ChooseMove(board, legal-moves) → best-move
  - Or could learn an ***evaluation function***, *V*(board) → $\mathcal{R}$, that gives each board position a score for how favorable it is. *V* can be used to pick a move by applying each legal move, scoring the resulting board position, and choosing the move that results in the highest scoring board position.

Ideal Definition of $V(b)$

- If b is a final winning board, then $V(b) = 100$
- If b is a final losing board, then $V(b) = -100$
- If b is a final draw board, then $V(b) = 0$
- Otherwise, then $V(b) = V(b')$, where b' is the highest scoring final board position that is achieved starting from b and playing optimally until the end of the game (assuming the opponent plays optimally as well).
  - Can be computed using complete mini-max search of the finite game tree.

Approximating *V*(*b*)

- Computing *V*(*b*) is intractable since it involves searching the complete exponential game tree.
- Therefore, this definition is said to be ***non-operational***.
- An ***operational*** definition can be computed in reasonable (polynomial) time.
- Need to learn an operational *approximation* to the ideal evaluation function.

Linear Function for Representing $V(b)$

- In checkers, use a linear approximation of the evaluation function.

$$\hat{V}(b) = w_0 + w_1 \cdot bp(b) + w_2 \cdot rp(b) + w_3 \cdot bk(b) + w_4 \cdot rk(b) + w_5 \cdot bt(b) + w_6 \cdot rt(b)$$

  - *bp*(*b*): number of black pieces on board b
  - *rp*(*b*): number of red pieces on board b
  - *bk*(*b*): number of black kings on board b
  - *rk*(*b*): number of red kings on board b
  - *bt*(*b*): number of black pieces threatened (i.e. which can be immediately taken by red on its next turn)
  - *rt*(*b*): number of red pieces threatened

Representing the Target Function

- Target function can be represented in many ways: lookup table, symbolic rules, numerical function, neural network.
- There is a trade-off between the expressiveness of a representation and the ease of learning.
- The more expressive a representation, the better it will be at approximating an arbitrary function; however, the more examples will be needed to learn an accurate function.

Obtaining Training Values

- Direct supervision may be available for the target function.
  - < <*bp*=3,*rp*=0,*bk*=1,*rk*=0,*bt*=0,*rt*=0>, 100> (win for black)
- With indirect feedback, training values can be estimated using ***temporal difference learning*** (used in ***reinforcement learning*** where supervision is ***delayed reward***).

Temporal Difference Learning

- Estimate training values for intermediate (non-terminal) board positions by the estimated value of their successor in an actual game trace. $$V_{train}(b) = \hat{V}(\mathrm{successor}(b))$$

  where successor(*b*) is the next board position where it is the program's move in actual play.
- Values towards the end of the game are initially more accurate and continued training slowly "backs up" accurate values to earlier board positions.

Learning Algorithm

- Looks for w0…w6
- Uses training values
- Attempts to minimize some measure of error (***loss function***) such as ***mean squared error***:

$$E = \frac{\sum_{b \in B} [V_{train}(b) - \hat{V}(b)]^2}{|B|}$$

A gradient descent algorithm

- A gradient descent algorithm that incrementally updates the weights of a linear function in an attempt to minimize the mean squared error

Until weights converge :

For each training example *b* do :

1) Compute the absolute error :

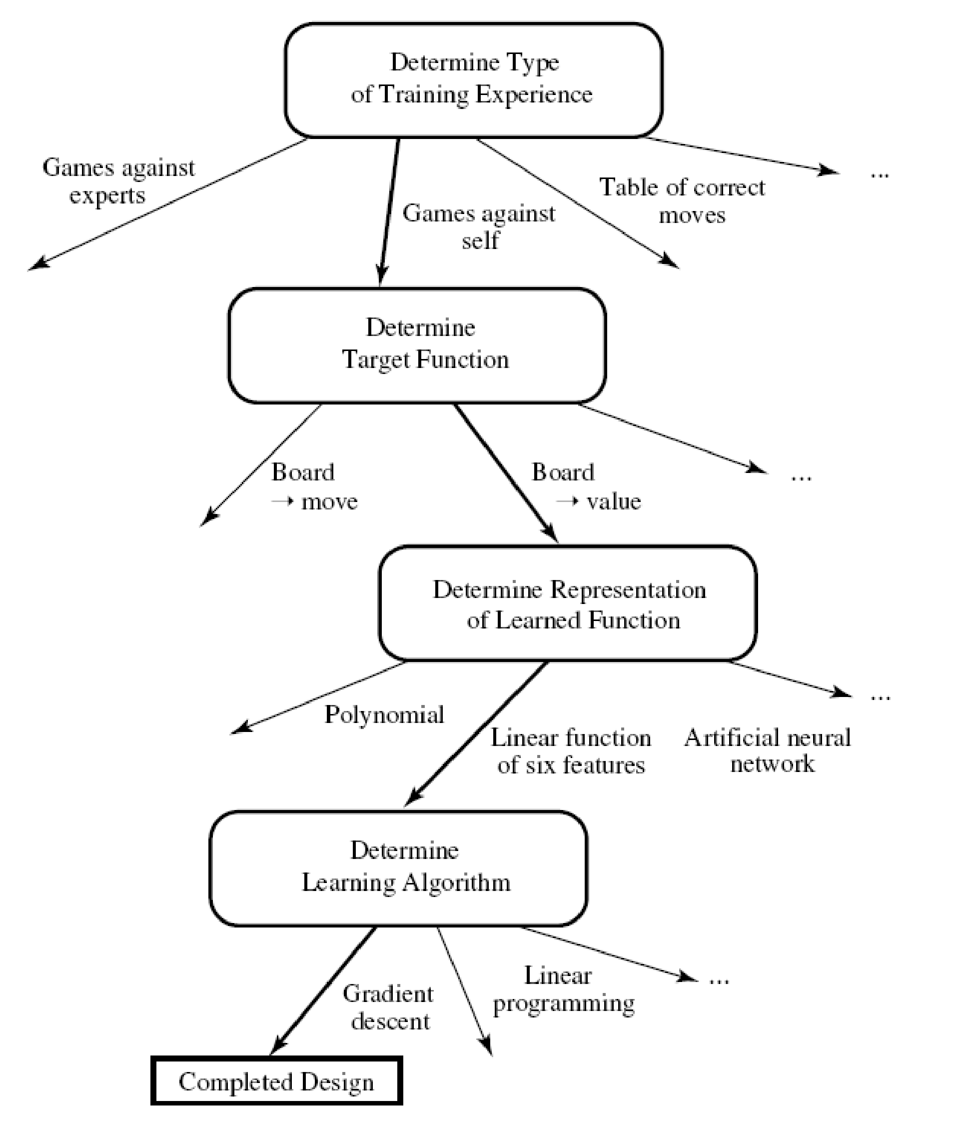
$$error(b) = V_{train}(b) - \widehat{V}(b)$$

2) For each board feature, f_i, update its weight, w_i :

$$w_i = w_i + c \cdot f_i \cdot error(b)$$

for some small constant (learning rate) c

Design Choice



Hypothesis Space

- One way to think about a supervised learning machine is as a device that explores a “hypothesis space”.
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.

- The art of supervised machine learning is in:
  - Deciding how to represent the inputs and outputs
  - Selecting a hypothesis space that is powerful enough to represent the relationship between inputs and outputs but simple enough to be searched.

Lessons Learned about Learning

- Learning can be viewed as using direct or indirect experience to approximate a chosen target function.
- Function approximation can be viewed as a search through a space of hypotheses (representations of functions) for one that best fits a set of training data.
- Different learning methods assume different hypothesis spaces (representation languages) and/or employ different search techniques.

Various Function Representations

- Numerical functions
  - Linear regression
  - Neural networks
  - Support vector machines
- Symbolic functions
  - Decision trees
  - Rules in propositional logic
  - Rules in first-order predicate logic
- Instance-based functions
  - Nearest-neighbor
  - Case-based
- Probabilistic Graphical Models
  - Naïve Bayes
  - Bayesian networks
  - Hidden-Markov Models (HMMs)
  - Probabilistic Context Free Grammars (PCFGs)
  - Markov networks

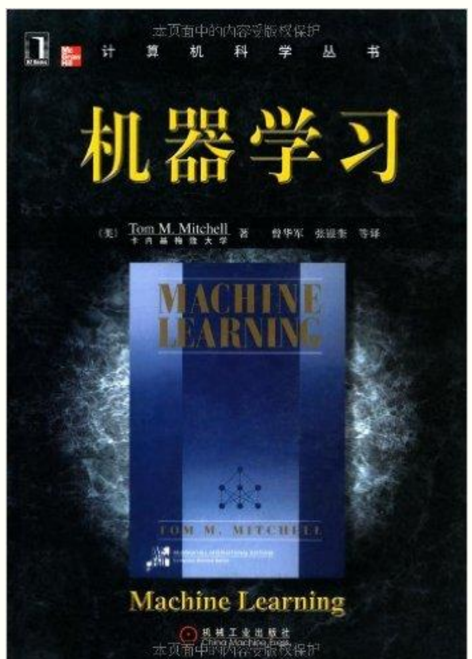
Various Search Algorithms

- Gradient descent
  - Perceptron
  - Backpropagation
- Dynamic Programming
  - HMM Learning
  - PCFG Learning
- Divide and Conquer
  - Decision tree induction
  - Rule learning
- Evolutionary Computation
  - Genetic Algorithms (GAs)
  - Genetic Programming (GP)
  - Neuro-evolution

Course Arrangement

- 1-12：
  - Concept learning;
  - Decision tree learning;
  - Linear models;
  - Artificial neural network;
  - Bayesian learning;
  - Instance based learning;
  - Genetic algorithms;
  - Application;
- 3： three course works;
- 16:Review
- 17：Final test

- Tom Mitchell. “Machine Learning”. McGraw-Hill, 1997



Goals of This Course

- Know about the big picture of machine learning
- Get familiar with popular ML methodologies
- Get some first-hand ML developing experiences
- Present your own ML solutions to real-world

problems

Resources: Datasets

- UCI Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html
- UCI KDD Archive: http://kdd.ics.uci.edu/summary.data.application.html
- Statlib: http://lib.stat.cmu.edu/
- Delve: http://www.cs.utoronto.ca/~delve/

Resources: Journals

- Journal of Machine Learning Research
  www.jmlr.org
- Machine Learning
- Neural Computation
- Neural Networks
- IEEE Transactions on Neural Networks
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- Annals of Statistics
- Journal of the American Statistical Association
- ...

Resources: Conferences

- International Conference on Machine Learning (ICML)
  - ICML05: http://icml.ais.fraunhofer.de/
- European Conference on Machine Learning (ECML)
  - ECML05: http://ecmlpkdd05.liacc.up.pt/
- Neural Information Processing Systems (NIPS)
  - NIPS05: http://nips.cc/
- Uncertainty in Artificial Intelligence (UAI)
  - UAI05: http://www.cs.toronto.edu/uai2005/
- Computational Learning Theory (COLT)
  - COLT05: http://learningtheory.org/colt2005/
- International Joint Conference on Artificial Intelligence (IJCAI)
  - IJCAI05: http://ijcai05.csd.abdn.ac.uk/
- International Conference on Neural Networks (Europe)
  - ICANN05: http://www.ibspan.waw.pl/ICANN-2005/
- ...