CS420 Machine Learning

Weinan Zhang
Shanghai Jiao Tong University
http://wnzhang.net

Spring Semester, 2018

http://wnzhang.net/teaching/cs420/index.html

Self Introduction – Weinan Zhang

- Position
  - Assistant Professor at John Hopcroft Center, CS Dept. of SJTU 2016-now
  - Apex Data and Knowledge Management Lab
  - Research on machine learning and data mining topics
- Education
  - Ph.D. on Computer Science from University College London (UCL), United Kingdom, 2012-2016
  - B.Eng. on Computer Science from ACM Class 07 of Shanghai Jiao Tong University, China, 2007-2011

Course Administration

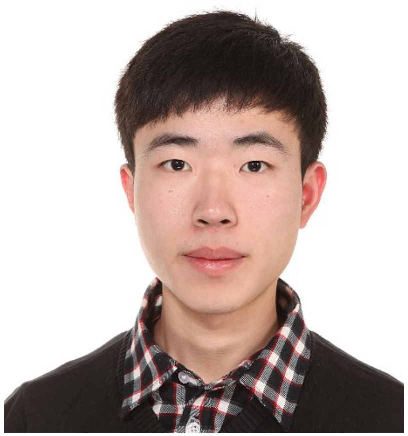
- No official text book for this course, some recommended books are
  - 李航《统计学习方法》清华大学出版社，2012.
  - 周志华《机器学习》清华大学出版社，2016.
  - Tom Mitchell. “Machine Learning”. McGraw-Hill, 1997
  - Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie. “The Elements of Statistical Learning”. Springer 2004.
  - Chris Bishop. “Pattern Recognition and Machine Learning”. Springer 2006.
  - Richard S. Sutton and Andrew G. Barto. “Reinforcement Learning: An Introduction”. MIT, 2012.

Course Administration

- A hands-on machine learning course
    - No assignment, no paper exam
    - Select two out of three course works (80%)
        - Kaggle-in-Class competitions on Classification (40%)
        - Kaggle-in-Class competitions on Recommendation (40%)
        - MAgent battle game competition (40%)
    - Poster session (10%)
    - Attending (10%)
        - Could be evaluated by quiz

Teaching Assistants

- Jiacheng Yang (杨嘉成)
- kipsora [A.T.] gmail.com
- ACM15 student, research intern in Apexlab
- Research on AutoML, reinforcement learning, deep learning
- Papers: NIPS & AAAI (published as demos), ICML (submitted)



- Lianmin Zheng (郑怜悯)
- mercy_zheng [A.T.] apex.sjtu.edu.cn
- ACM15 student, research intern in Apexlab
- Research on machine learning systems, multi-agent reinforcement learning
- Papers: NIPS & AAAI (published as demos)

Apexlab website: http://apex.sjtu.edu.cn/

TA Administration

- Join the mail list
  - Please send your
    - Chinese name
    - Student number
    - Email address

    to mercy_zheng [A.T] sjtu.edu.cn

    with email title “Check in CS420 2018”

- Office hour
  - Every Wednesday 7-8pm, 307 Yifu Building
  - TAs will be there for QA

Goals of This Course

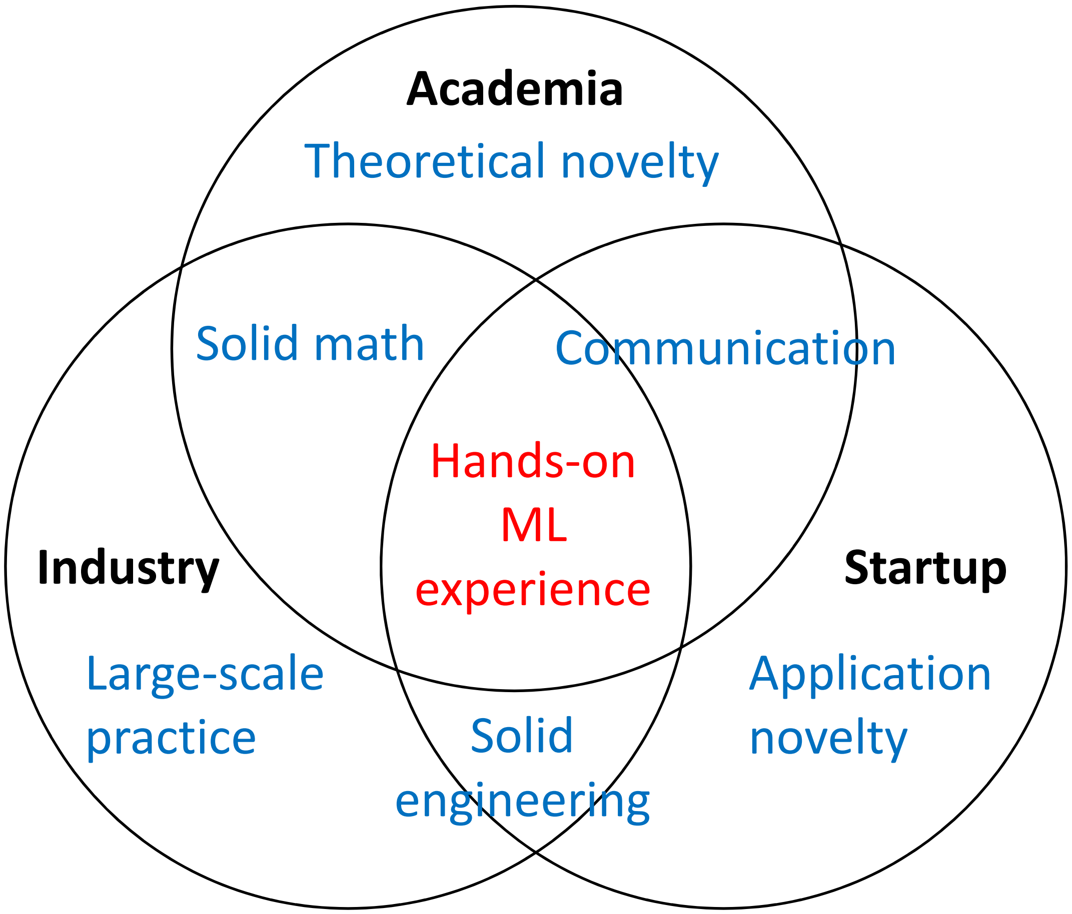
- Know about the big picture of machine learning

- Get familiar with popular ML methodologies
  - Data representations
  - Models
  - Learning algorithms
  - Experimental methodologies

- Get some first-hand ML developing experiences

- Present your own ML solutions to real-world problems

Why we focus on hands-on ML



- So play with the data and get your hands dirty!

Course Landscape

1. ML Introduction
2. Linear Models
3. SVMs and Kernels [cw1]
4. Neural Networks
5. Tree Models
6. Ensemble Models
7. Ranking and Filtering [cw2]
8. Graphic Models
9. Unsupervised Learning
10. Model Selection
11. RL Introduction [cw3]
12. Model-free RL
13. Multi-agent RL
14. Transfer Learning
15. Advanced ML
16. Poster Session

Introduction to Machine Learning

Weinan Zhang

Shanghai Jiao Tong University

http://wnzhang.net

http://wnzhang.net/teaching/cs420/index.html

Artificial Intelligence

- Artificial intelligence (AI) is intelligence exhibited by machines.
- The subject AI is about the methodology of designing machines to accomplish intelligence-based tasks.
- Intelligence is the computational part of the ability to achieve goals in the world.

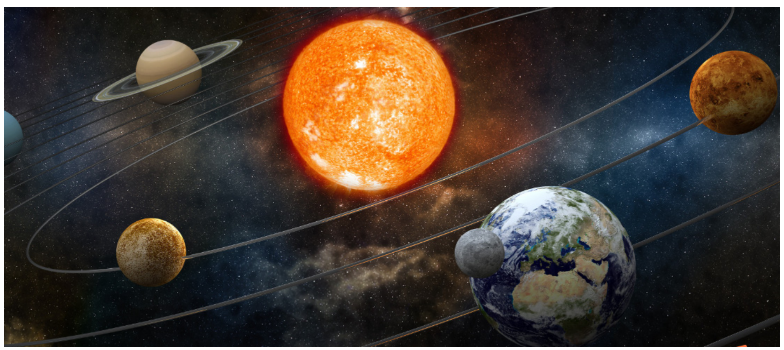
http://www-formal.stanford.edu/jmc/whatisai/whatisai.html

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

What is Data Science

- Physics
  - Goal: discover the underlying principle of the world



  - Solution: build the model of the world from observations

$$F = G\frac{m_1 m_2}{r^2}$$

- Data Science
  - Goal: discover the underlying principle of the data

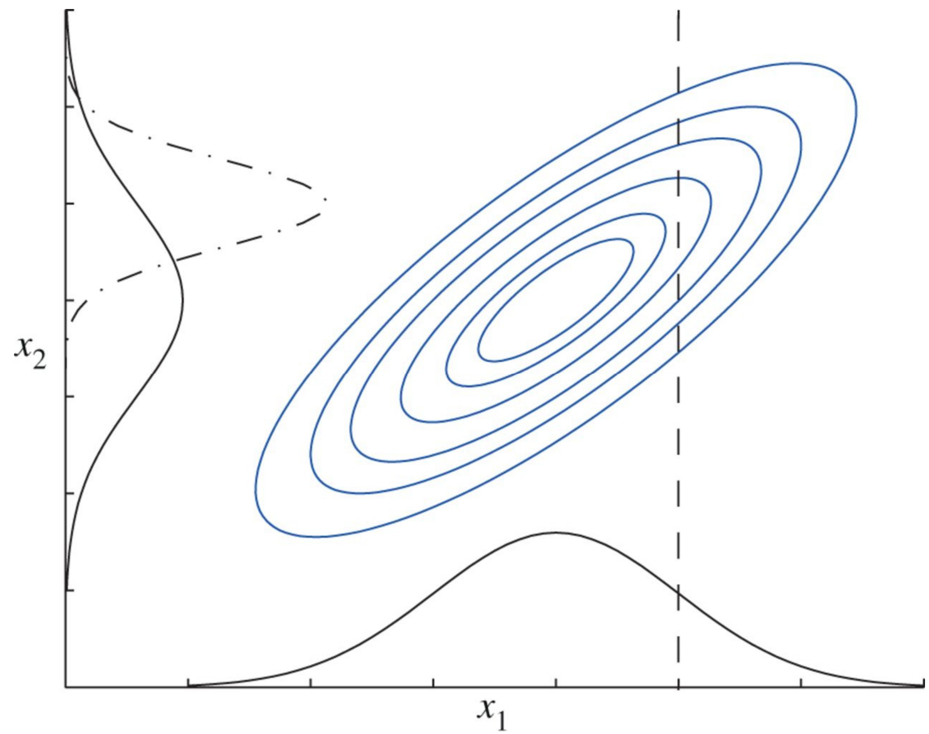
  - Solution: build the model of the data from observations

$$p(x) = \frac{e^{f(x)}}{\sum_{x'} e^{f(x')}}$$

Data Science

- Mathematically
  - Find joint data distribution $p(x)$
  - Then the conditional distribution $p(x_2|x_1)$
- Gaussian distribution
  - Multivariate

$$p(x) = \frac{e^{-(x-\mu)^\top \Sigma^{-1}(x-\mu)}}{\sqrt{|2\pi\Sigma|}}$$



  - Univariate

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

A Simple Example in User Behavior Modelling

Interest	Gender	Age	BBC Sports	PubMed	Bloomberg Business	Spotify
Finance	Male	29	Yes	No	Yes	No
Sports	Male	21	Yes	No	No	Yes
Medicine	Female	32	No	Yes	No	No
Music	Female	25	No	No	No	Yes
Medicine	Male	40	Yes	Yes	Yes	No

- Joint data distribution

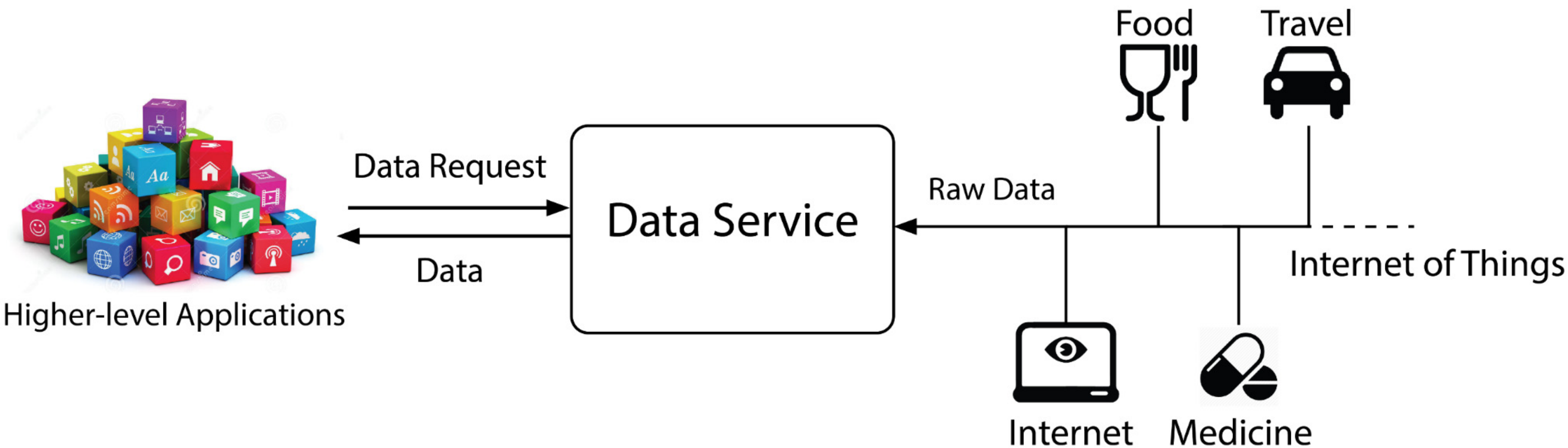
p(Interest=Finance, Gender=Male, Age=29, Browsing=BBC Sports,Bloomberg Business)

- Conditional data distribution

p(Interest=Finance | Browsing=BBC Sports,Bloomberg Business)

p(Gender=Male | Browsing=BBC Sports,Bloomberg Business)

Data Technology



Data itself is not valuable, data service is!

What is Machine Learning

- Learning

“Learning is any process by which a system improves performance from experience.”

--- Herbert Simon
Carnegie Mellon University

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

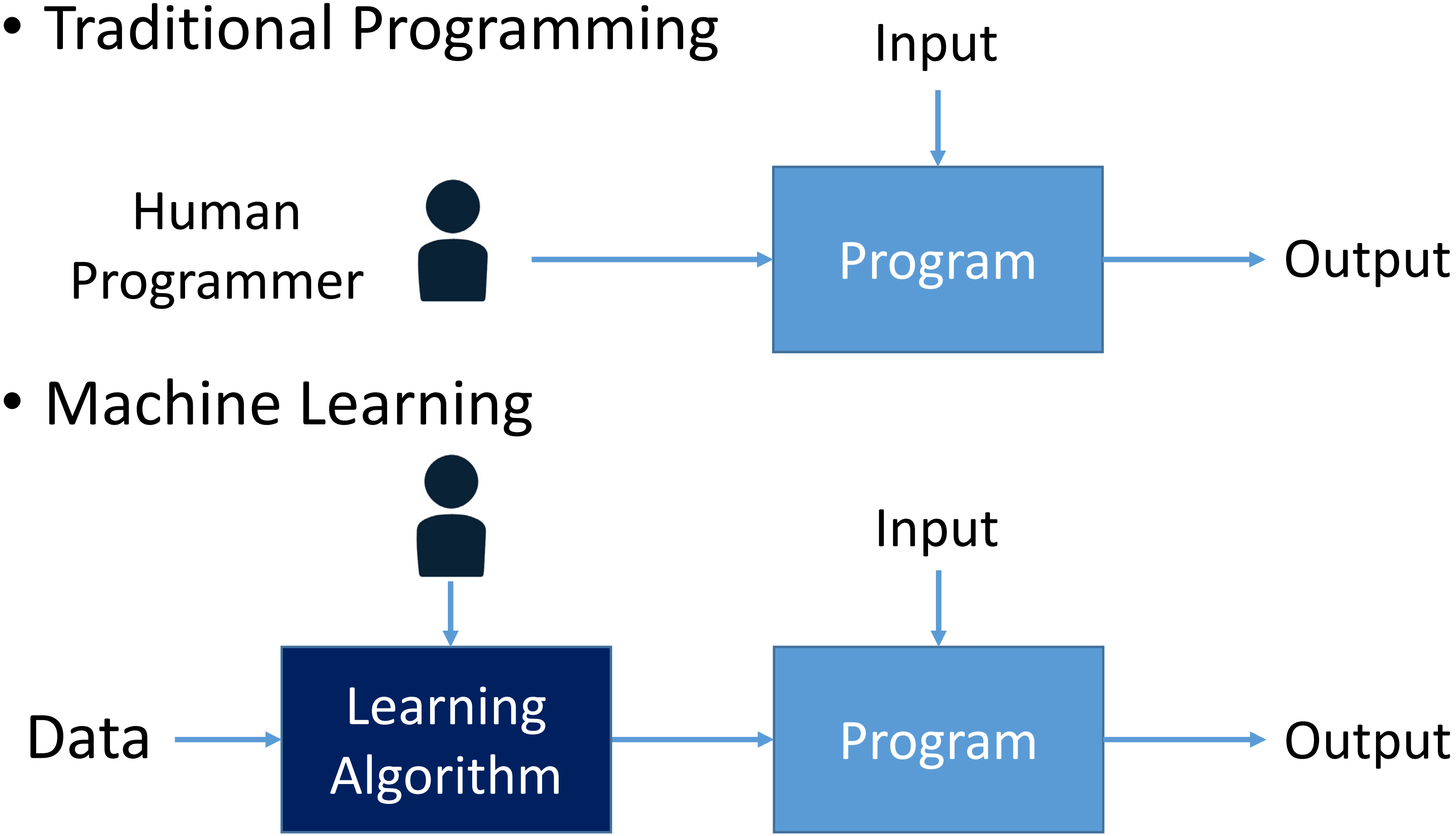
A more mathematical definition by Tom Mitchell

- Machine learning is the study of algorithms that
  - improve 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*>

Programming vs. Machine Learning

- Traditional Programming
- Machine Learning



Slide credit: Feifei Li

When does ML Make Advantages

ML is used when

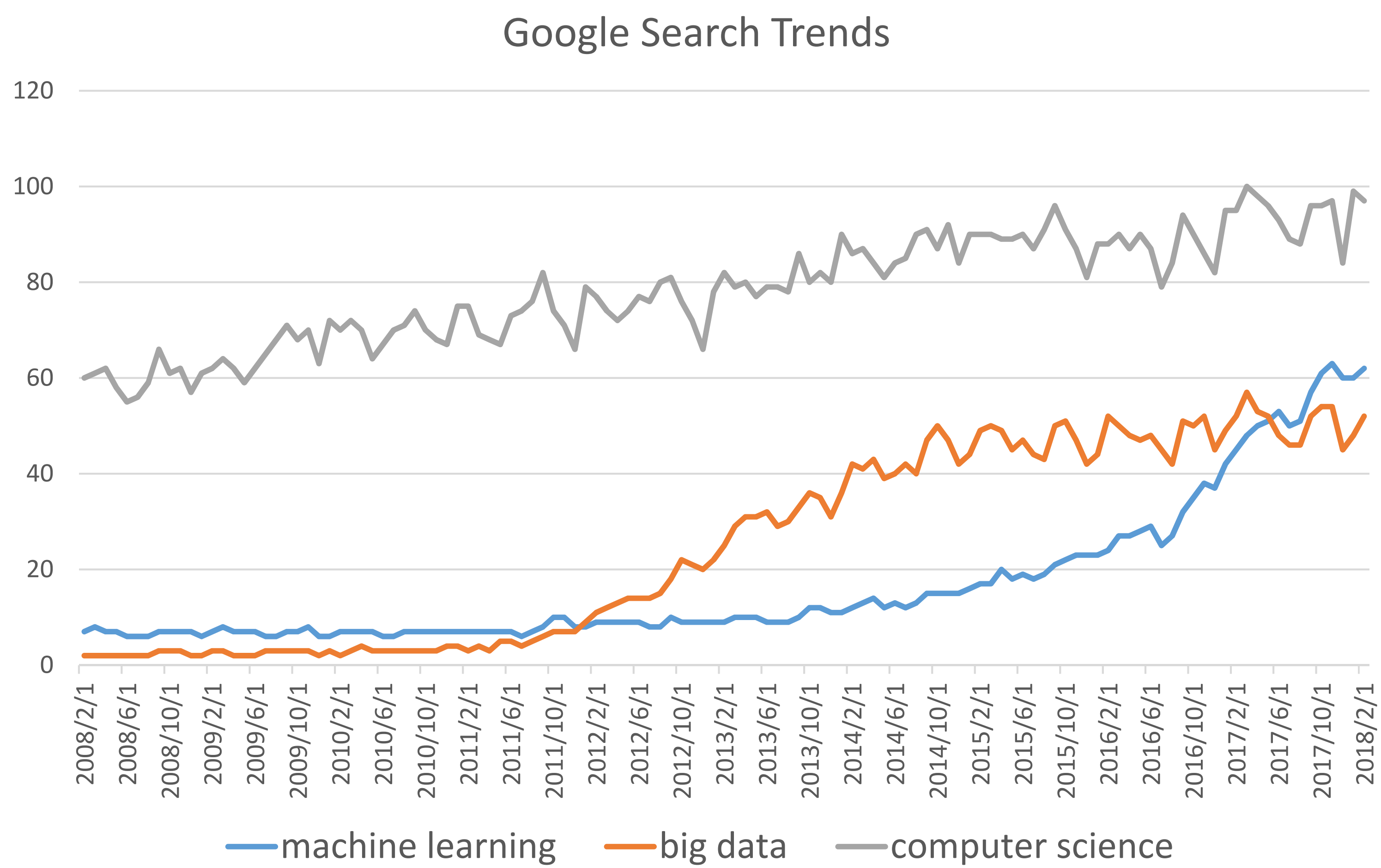
- Models are based on a huge amount of data
  - Examples: Google web search, Facebook news feed
- Output must be customized
  - Examples: News / item / ads recommendation
- Humans cannot explain the expertise
  - Examples: Speech / face recognition, game of Go
- Human expertise does not exist
  - Examples: Navigating on Mars

Two Kinds of Machine Learning

- Prediction
  - Predict the desired output given the data (supervised learning)
  - Generate data instances (unsupervised learning)

- Decision Making
  - Take actions in a dynamic environment (reinforcement learning)
    - to transit to new states
    - to receive immediate reward
    - to maximize the accumulative reward over time

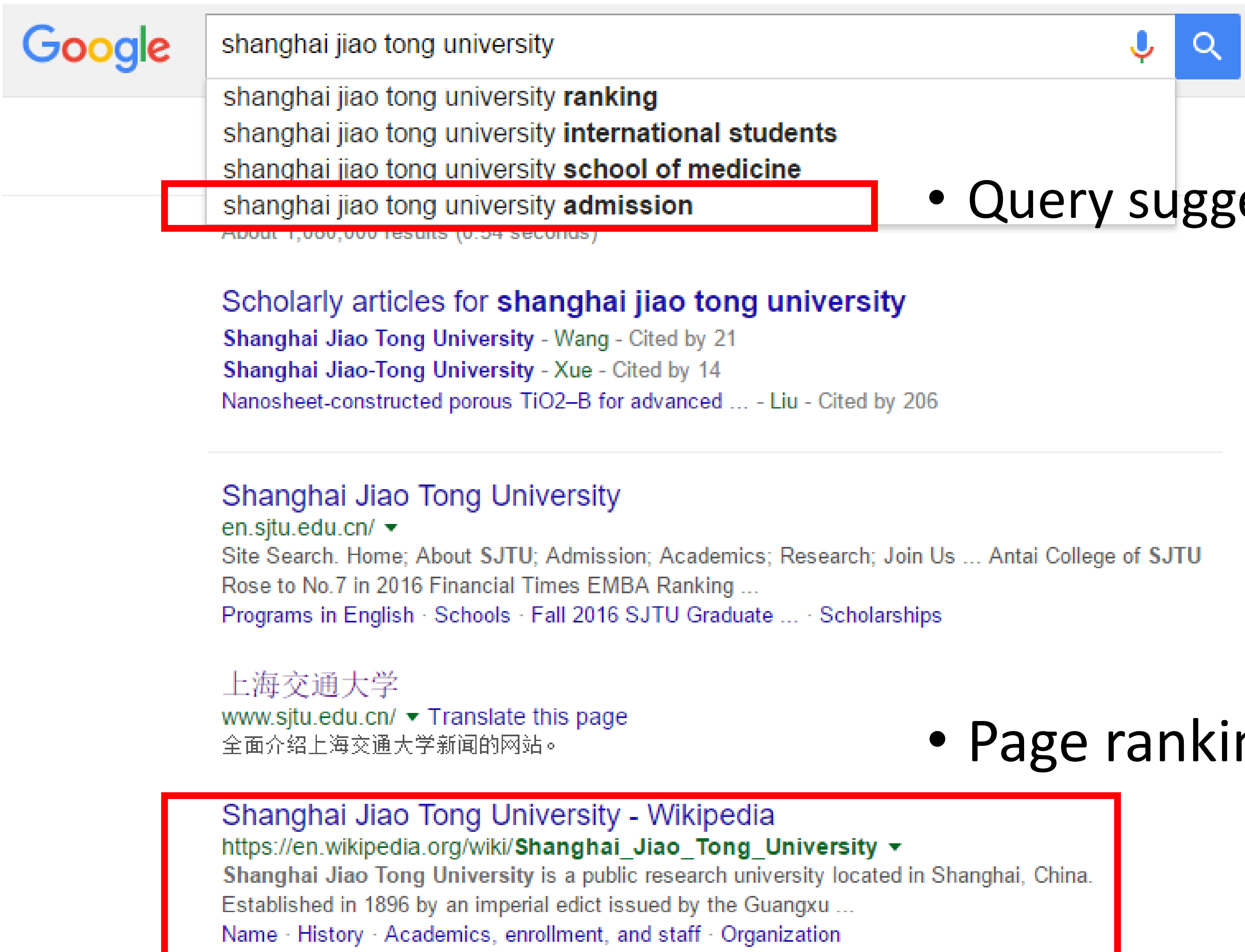
Trends



https://www.google.com/trends

Some ML Use Cases

ML Use Case 1: Web Search

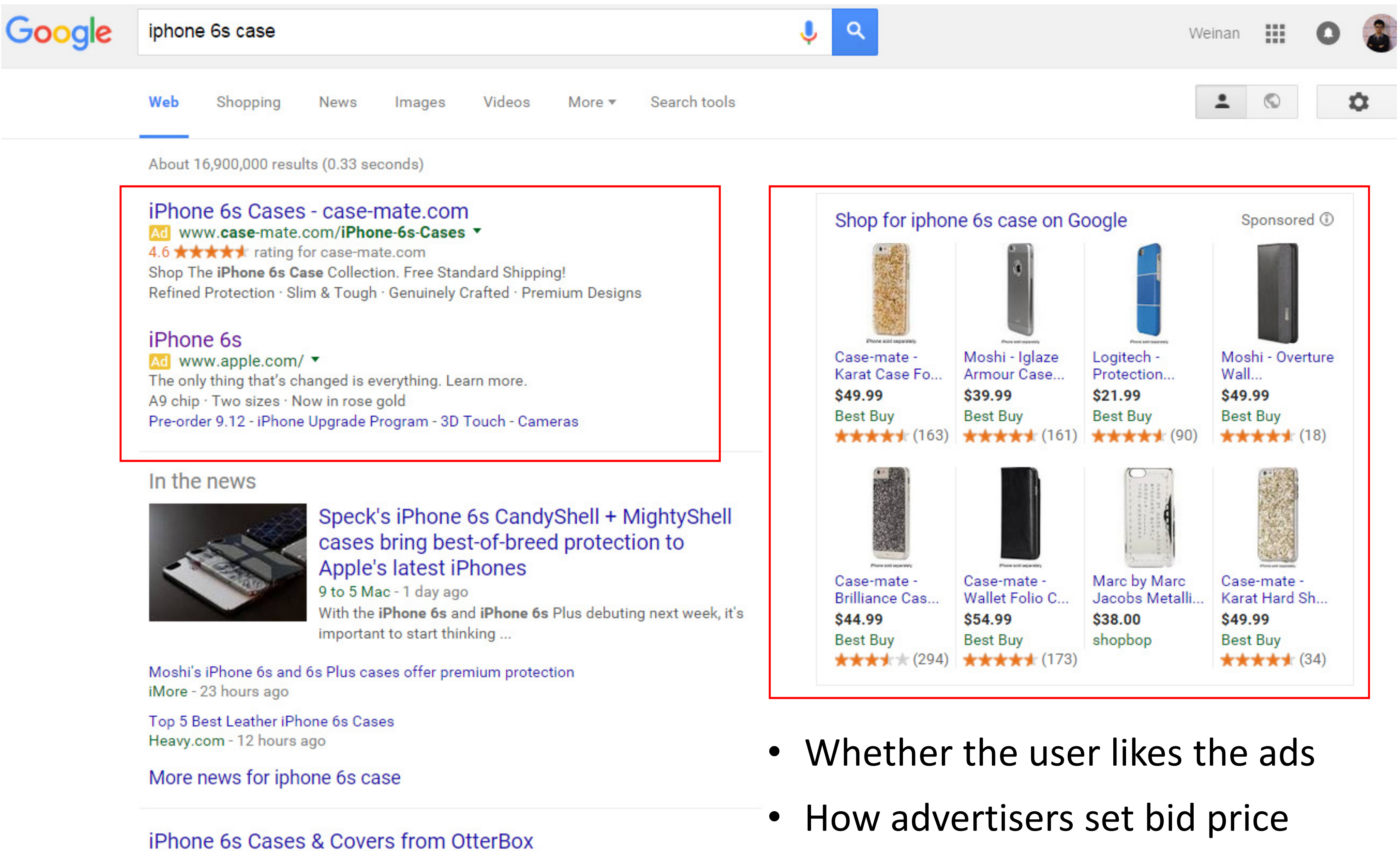


- Query suggestion
- Page ranking

ML Use Case 2: News Recommendation

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

ML Use Case 3: Online Advertising



- Whether the user likes the ads
- How advertisers set bid price

ML Use Case 3: Online Advertising

- Whether the user likes the ads
- How advertisers set bid price

https://github.com/wnzhang/rtb-papers

ML Use Case 4: Information Extraction

Kinect - Fastest Selling Electronic Product in History

Posted on: 3/10/2011 1:09:45 PM by **David Lewis**

Microsoft's Kinect sensor system has been officially recognised as the fastest selling electrical device in history.

Manufactured to give wireless interactivity with the company's Xbox game platform, the device has sold eight million units in its first two months, outstripping the sales of Apple's iPhone and iPad when they were launched.

The news comes as a welcome relief for Microsoft who have been trailing Apple in the technology stakes over the last few years with the Apple brand being seen as more cool and sexy than Microsoft.

The figures, which have been verified by the Guinness Book of World Records, represent sales of the camera add-on which uses infrared technology to track the movement of the participant and translate their movements to action in the game.

For some time Microsoft's Xbox was at a disadvantage to Nintendo's Wii system because of the lack of a motion detector but the Kinect addresses the issue well. Microsoft were keen on using a different technological base for their system to avoid being accused of copyright infringement and so the solution was built around infrared technology.

Microsoft says that sales of the Kinect reflect the popularity of the games platform in comparison with the Wii and hope that the availability of Kinect will also boost sales of the Xbox itself.

It notes that sales of games for the Xbox have also rocketed since the device became available with total sales now exceeding ten million.

In January Microsoft reported profits of $6.63bn (£4.1bn) for the last three months of 2010, down from $6.66bn a year earlier despite the excellent sales performance of Kinect.

Posted: 3/10/2011 1:09:45 PM by **David Lewis** | with 0 comments

Webpage

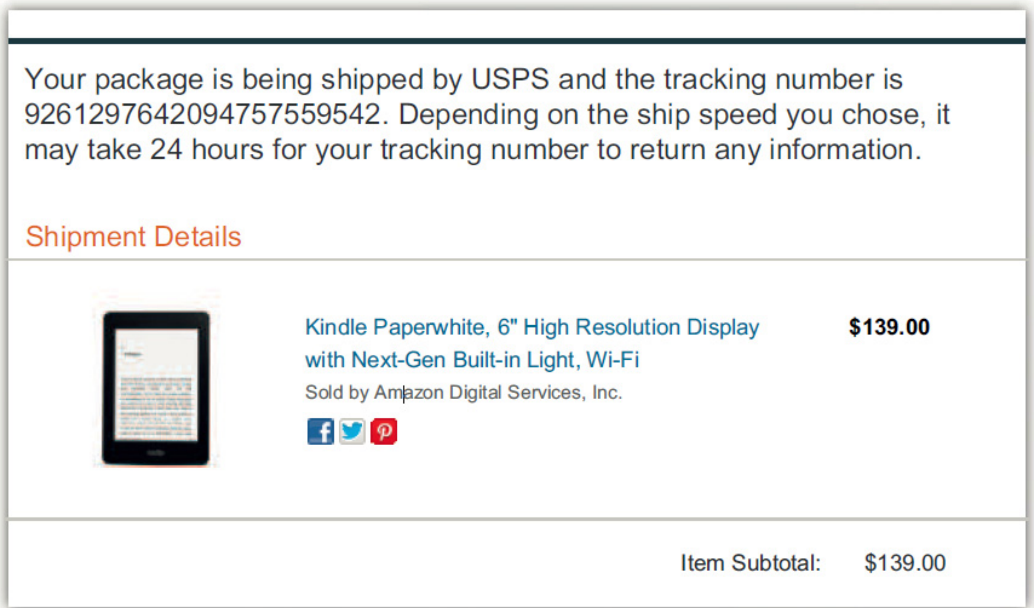
Kinect
Electronic Product
Microsoft's Xbox
Games
Xbox Game Platform

.
.
.

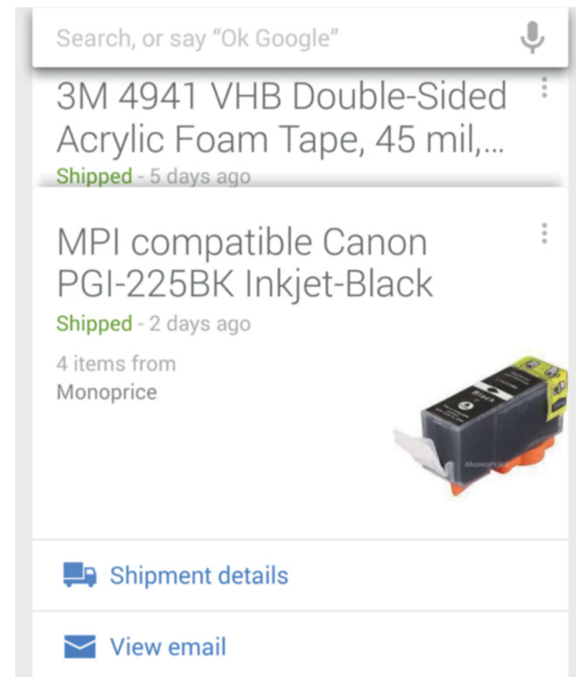
Keywords

ML Use Case 4: Information Extraction

- Structural information extraction and illustration



Gmail

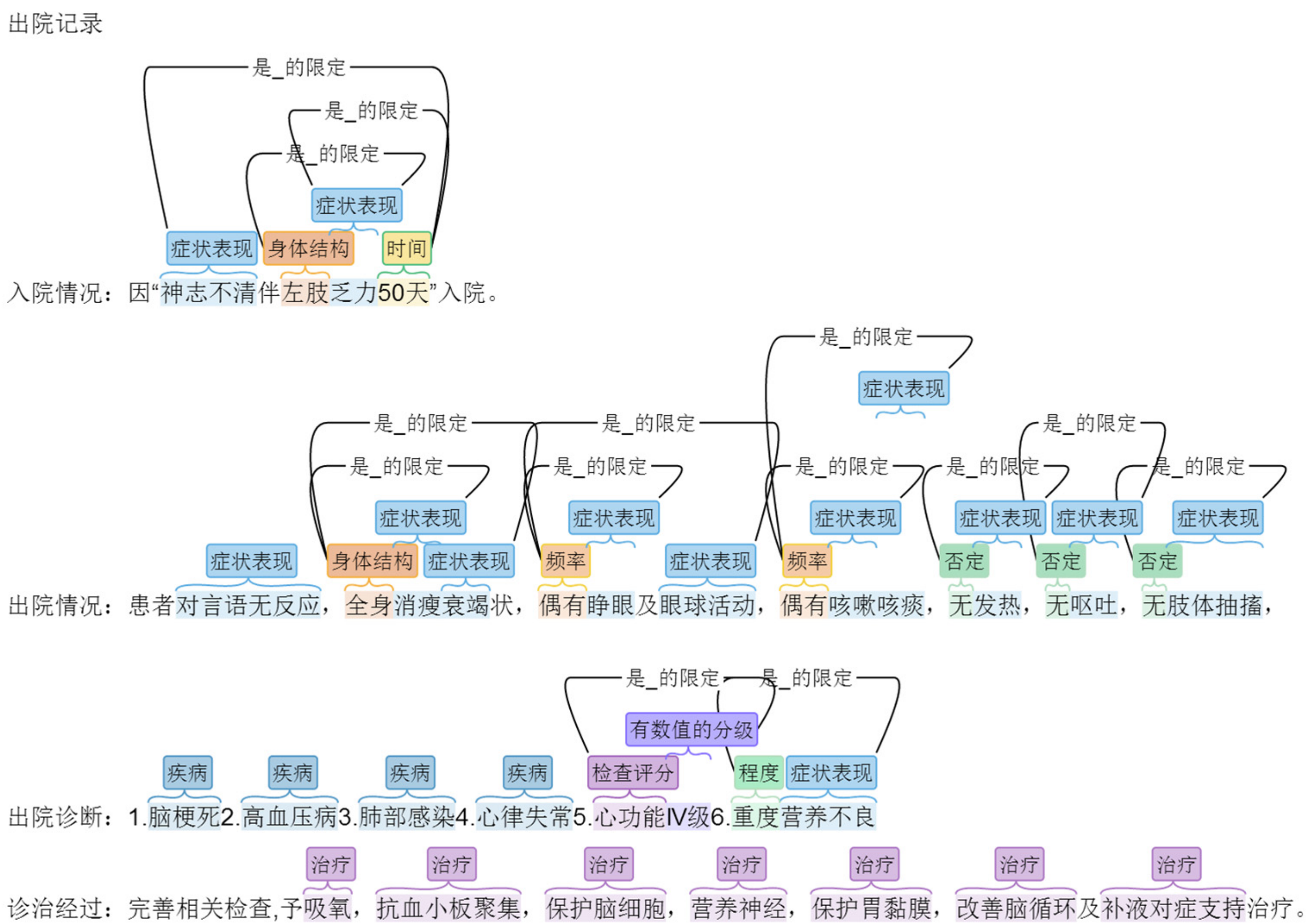


Google Now

Zhang, Weinan, et al. Annotating needles in the haystack without looking: Product information extraction from emails. KDD 2015.

ML Use Case 4: Information Extraction

- Clinical medicine structural information extraction

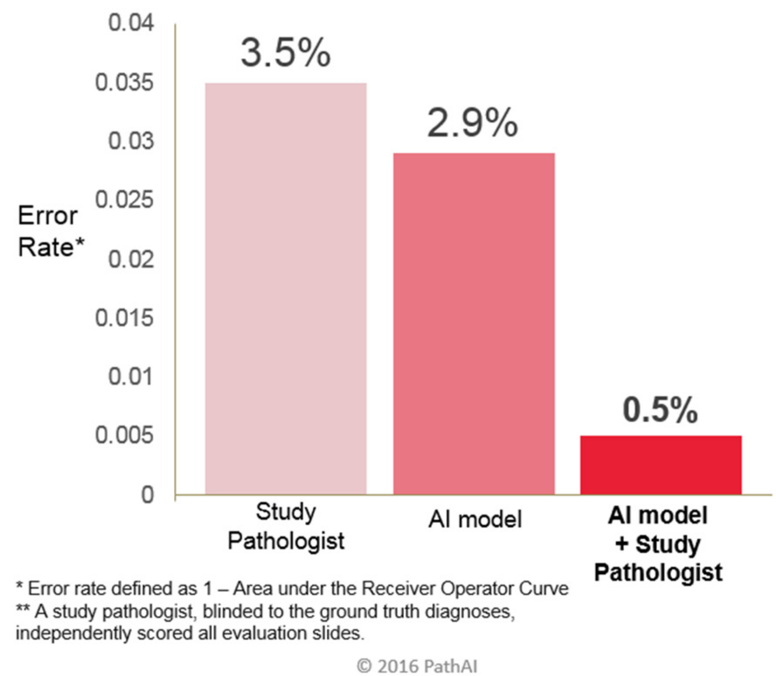


Zhenghui Wang, Weinan Zhang et al. Label-aware Double Transfer Learning for Cross Specialty Medical Named Entity Recognition. NAACL 2018.

ML Use Case 5: Medical Image Analysis

- Breast Cancer Diagnoses

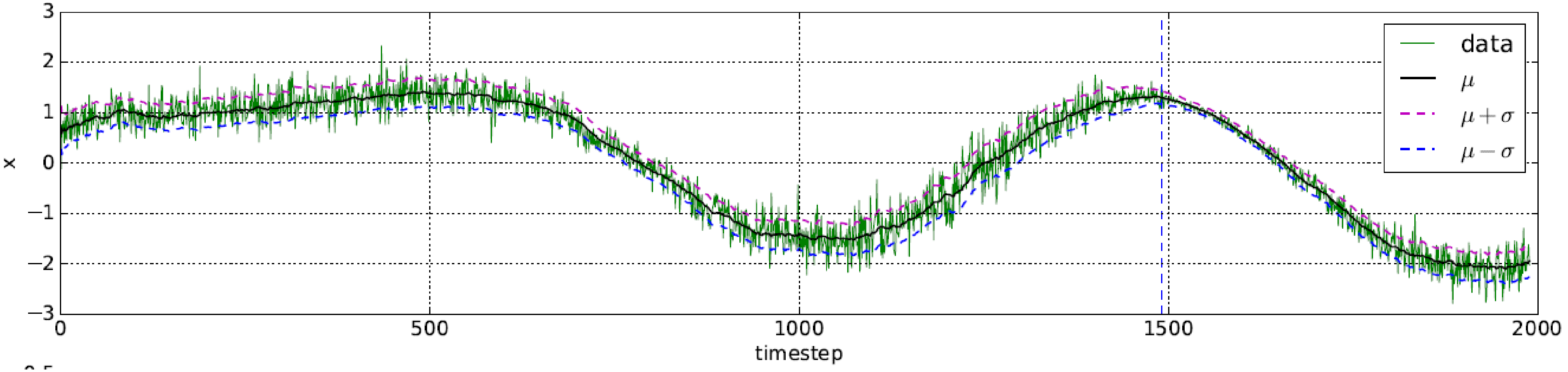
(AI + Pathologist) > Pathologist

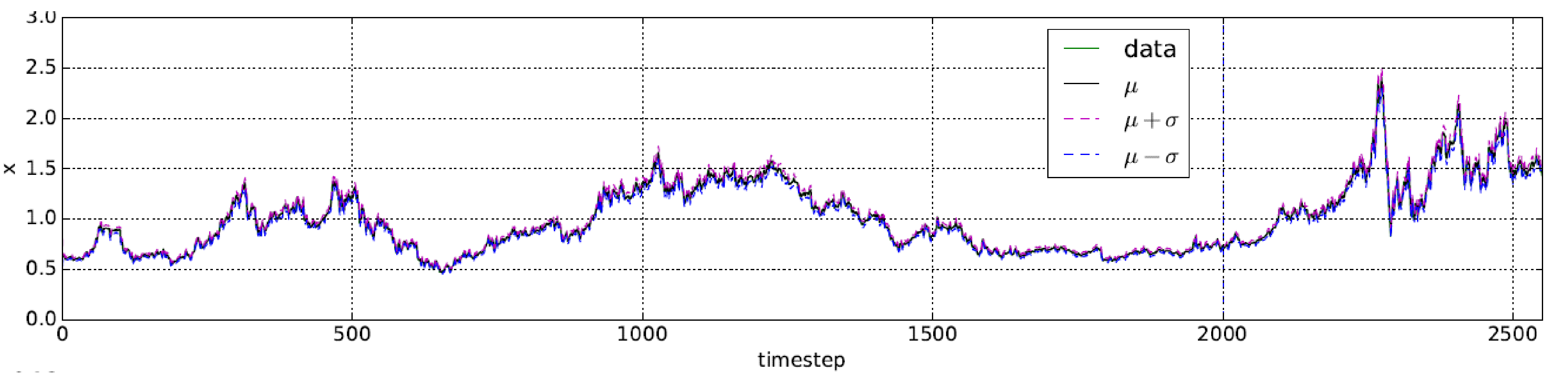


Wang, Dayong, et al. "Deep learning for identifying metastatic breast cancer." arXiv preprint arXiv:1606.05718 (2016).
https://blogs.nvidia.com/blog/2016/09/19/deep-learning-breast-cancer-diagnosis/

ML Use Case 6: Financial Data Prediction

- Predict the trend and volatility of financial data





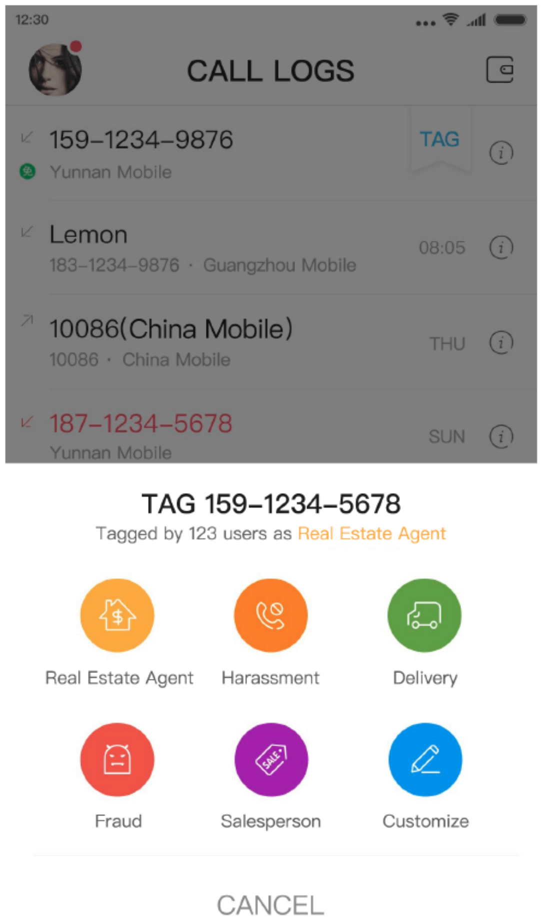
Rui Luo, Xiaojun Xu, Weinan Zhang et al. A Neural Stochastic Volatility Model. AAAI 2018.

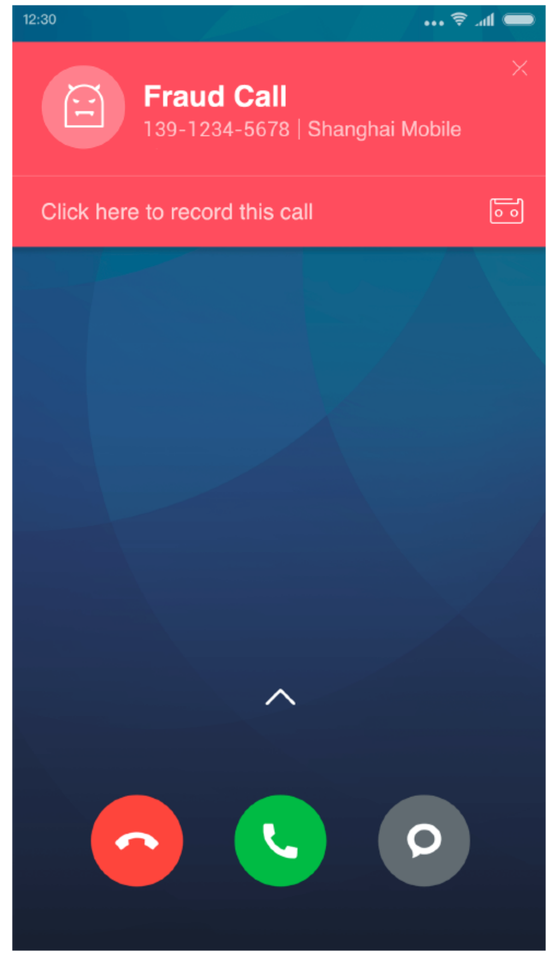
ML Use Case 7: Social Networks

- Friends/Tweets/Job Candidates suggestion

ML Use Case 8: Anomaly Detection

- Detect malicious calls

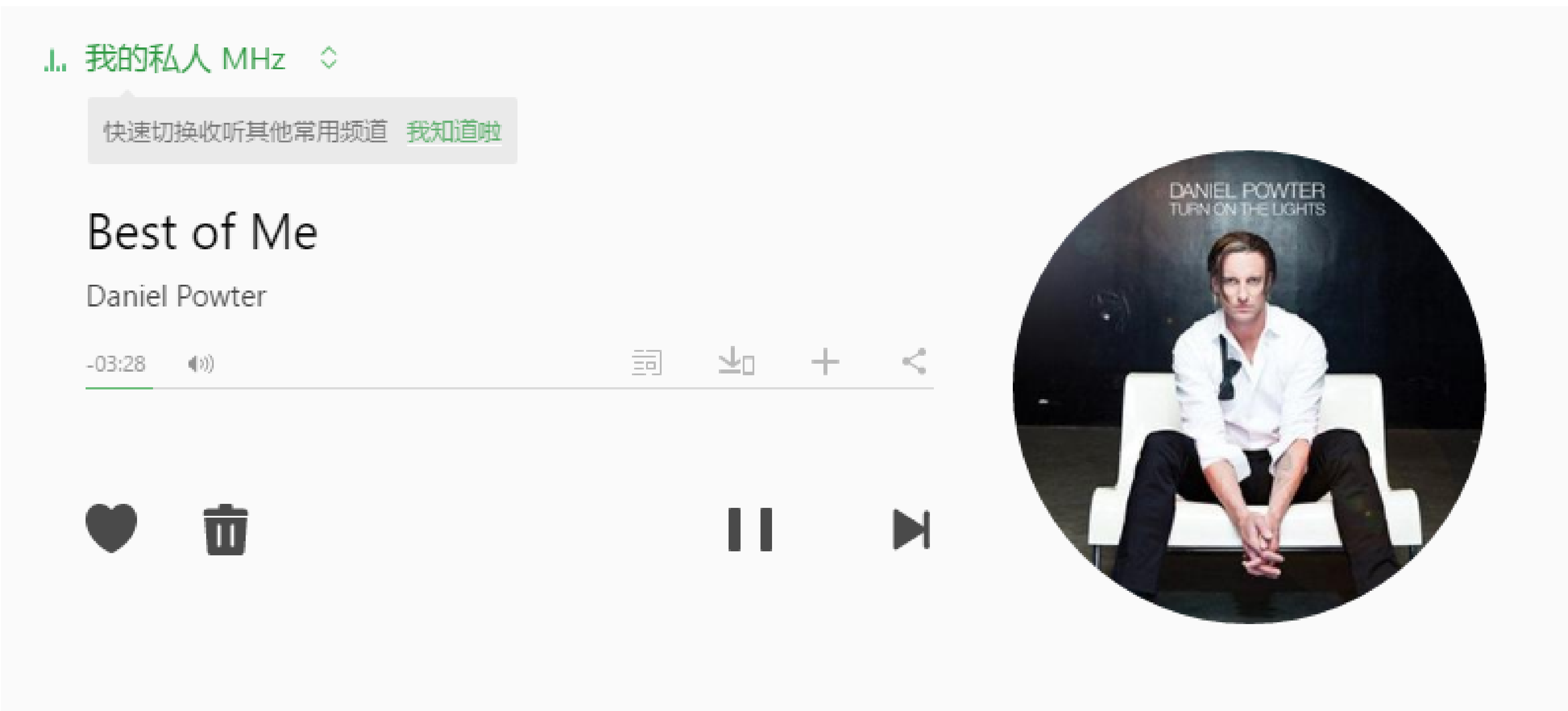




Huichen Li, Xiaojun Xu, Weinan Zhang et al. A Machine Learning Approach To Prevent Malicious Calls Over Telephony Networks. Oakland 2018.

ML Use Case 9: Interactive Recommendation

- Douban.fm music recommend and feedback
  - The machine needs to make decisions, not just prediction



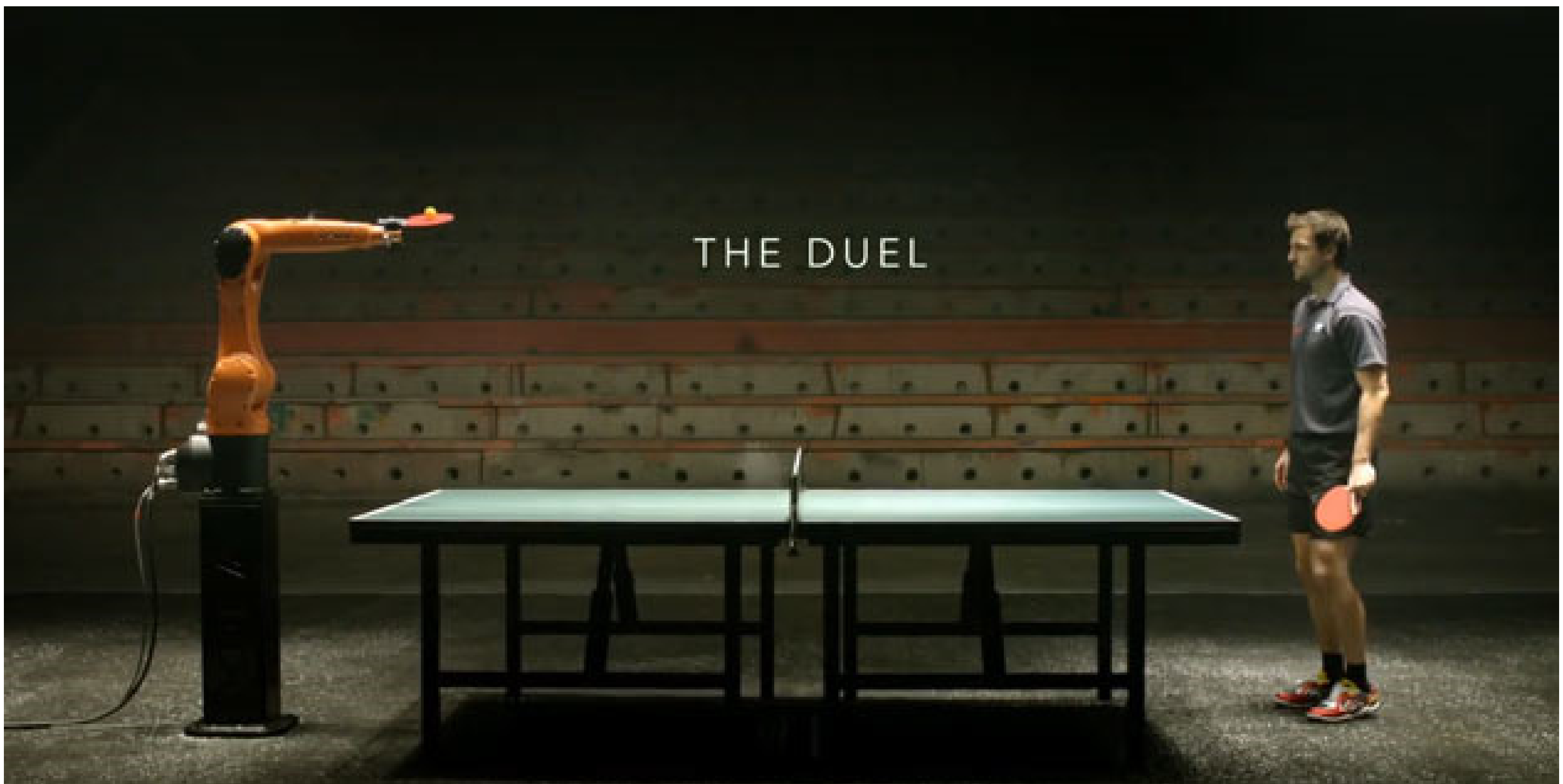
Xiaoxue Zhao, Weinan Zhang et al. Interactive Collaborative Filtering. CIKM 2013.

ML Use Case 10: Robotics Control

- Stanford Autonomous Helicopter
  - http://heli.stanford.edu/

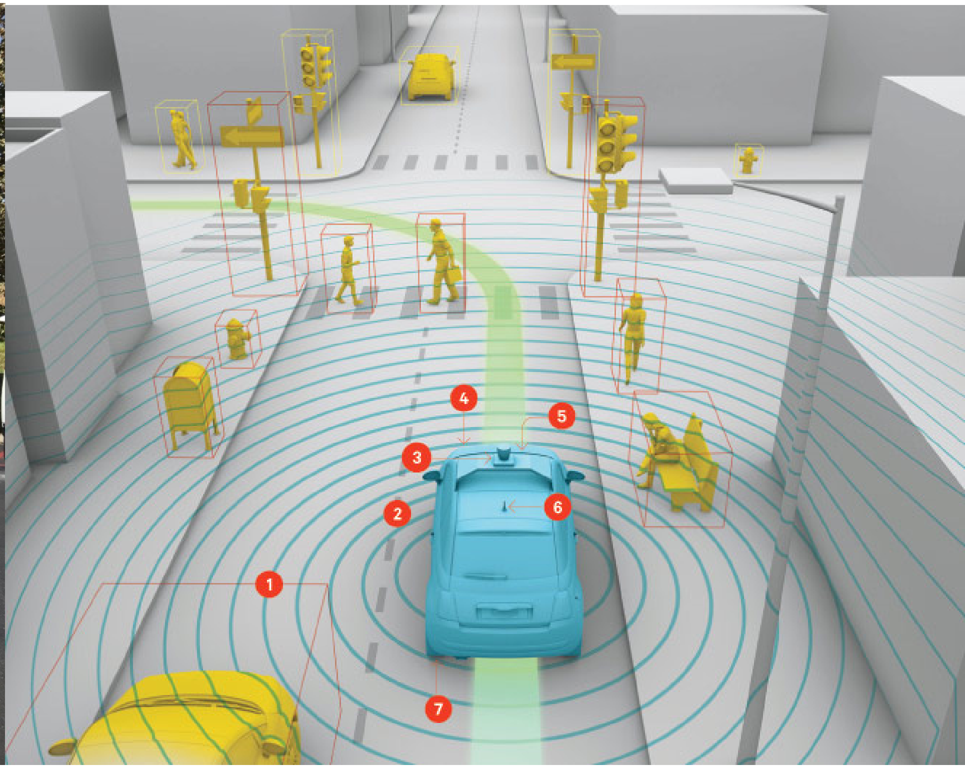
ML Use Case 10: Robotics Control

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



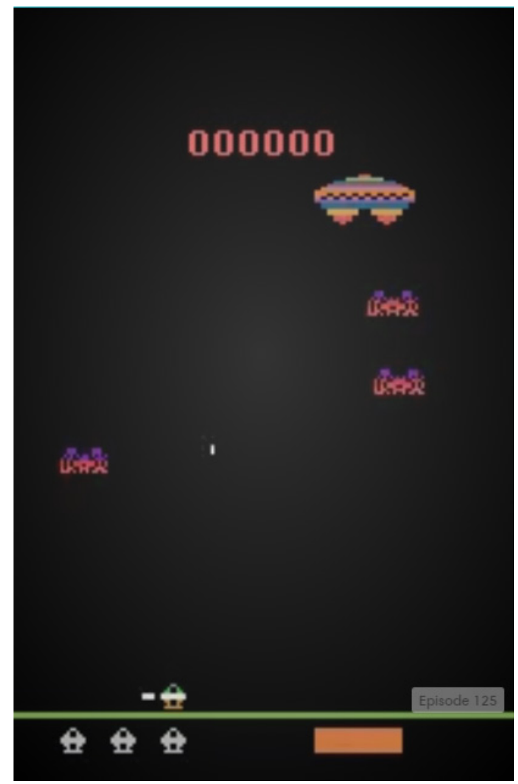
ML Use Case 11: Self-Driving Cars

- Google Self-Driving Cars
  - https://www.google.com/selfdrivingcar/



ML Use Case 12: Game Playing

- Take actions given screen pixels
  - https://gym.openai.com/envs#atari



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *Nature* 518.7540 (2015): 529-533.

ML Use Case 13: AlphaGo

IBM Deep Blue (1996)
- 4-2 Garry Kasparov on Chess
- A large number of crafted rules
- Huge space search

Google AlphaGo (2016)
- 4-1 Lee Sedol on Go
- Deep machine learning on big data

Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." *Nature* 529.7587 (2016): 484-489.

ML Use Case 14: Text Generation

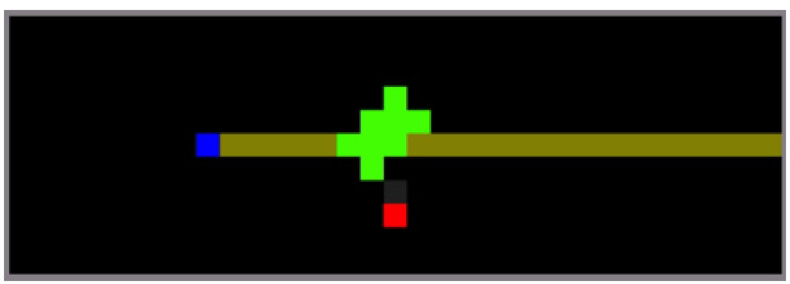
- Making decision of selecting the next word/char
- Chinese poem example. Can you distinguish?

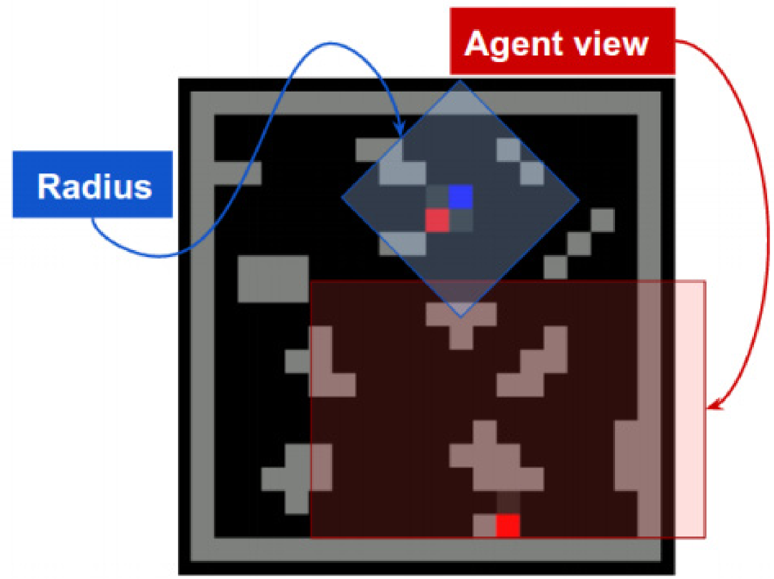
Human	Machine
南陌春风早，东邻去日斜。	山夜有雪寒，桂里逢客时。
紫陌追随日，青门相见时。	此时人且饮，酒愁一节梦。
胡风不开花，四气多作雪。	四面客归路，桂花开青竹。

Jiaxian Guo, Sidi Lu, Weinan Zhang et al. Long Text Generation via Adversarial Training with Leaked Information. AAAI 2018.
Lantao Yu, Weinan Zhang, et al. Seqgan: sequence generative adversarial nets with policy gradient. AAAI 2017.

ML Use Case 15: Multi-Agent Game Playing

- Multi-agent game playing
  - Learning to cooperate and compete





Leibo, Joel Z., et al. "Multi-agent Reinforcement Learning in Sequential Social Dilemmas." *AAMAS 2017.*

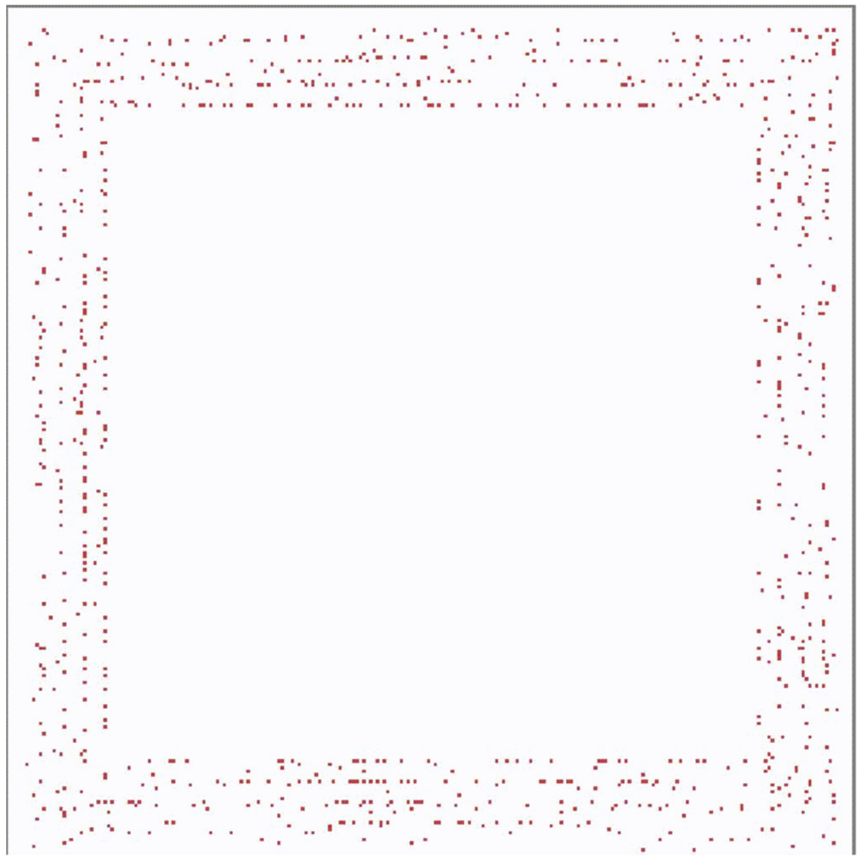
ML Use Case 15: Multi-Agent Game Playing

- Multi-agent game playing
  - Learning to cooperate and compete

Peng Peng, Jun Wang et al. Multiagent Bidirectionally-Coordinated Nets: Emergence of Human-level Coordination in Learning to Play StarCraft Combat Games. NIPS workshop 2017.

ML Use Case 16: Many-Agent Interactions

- MAgent game: aligning



Lianmin Zheng, Jiacheng Yang et al. MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence. NIPS 2017 & AAAI 2018 Demos.

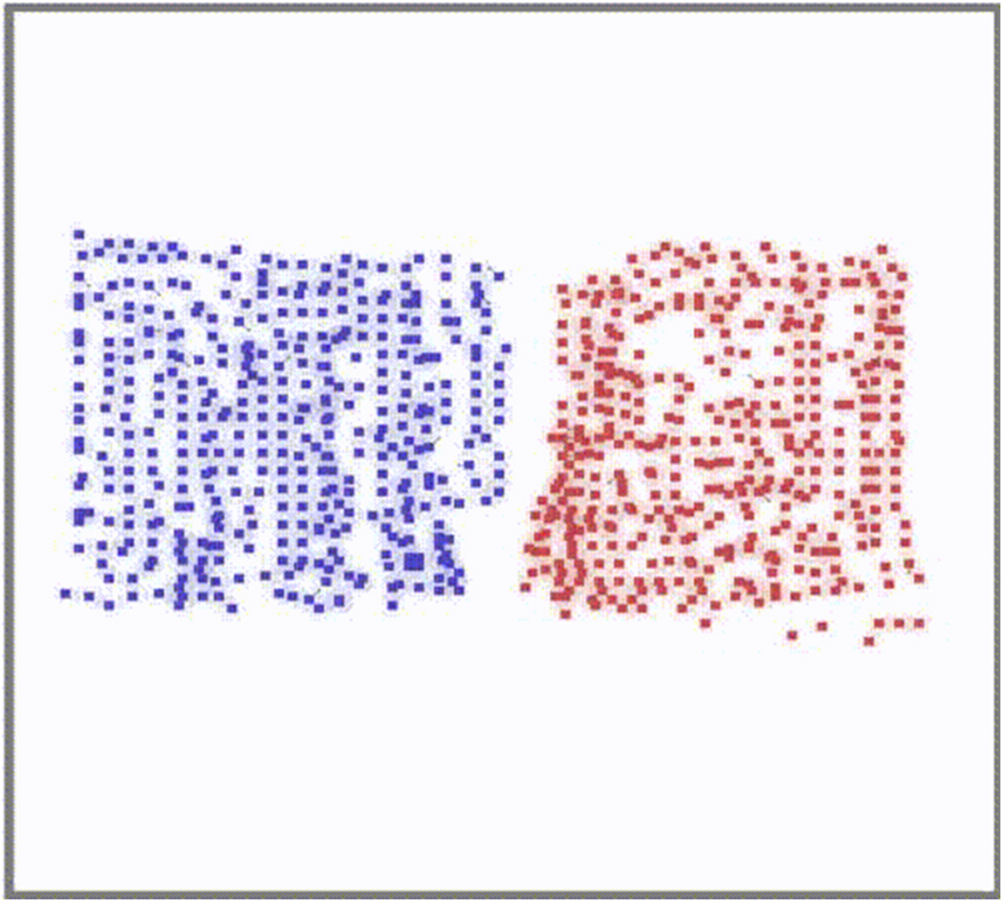
ML Use Case 16: Many-Agent Interactions

- MAgent game: city simulation

Lianmin Zheng, Jiacheng Yang et al. MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence. NIPS 2017 & AAAI 2018 Demos.

ML Use Case 16: Many-Agent Interactions

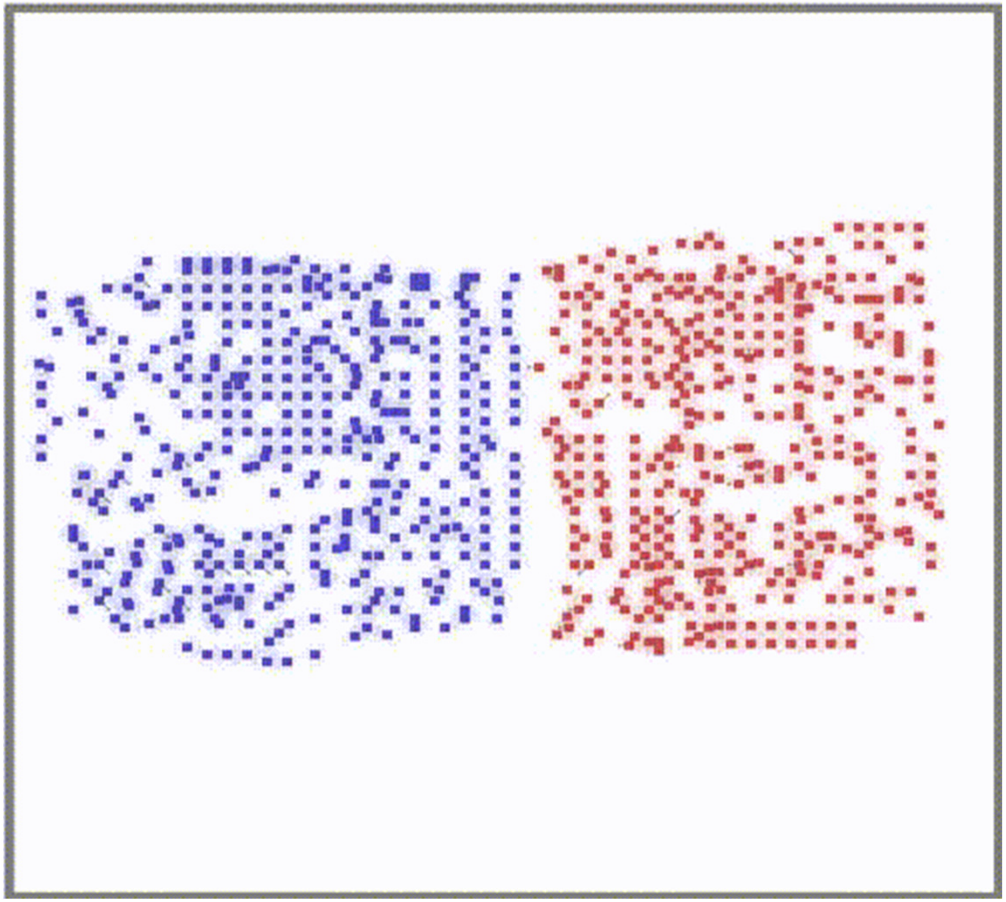
- MAgent game: battle



Lianmin Zheng, Jiacheng Yang et al. MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence. NIPS 2017 & AAAI 2018 Demos.

ML Use Case 16: Many-Agent Interactions

- MAgent game: battle



Lianmin Zheng, Jiacheng Yang et al. MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence. NIPS 2017 & AAAI 2018 Demos.

History of Machine Learning

- 1950s
  - Samuel's checker player
  - Machine learning term created
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - 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

Arthur Samuel coined the term "machine learning" in 1959

Slide credit: Ray Mooney

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

Slide credit: Ray Mooney

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

Slide credit: Ray Mooney

History of Machine Learning

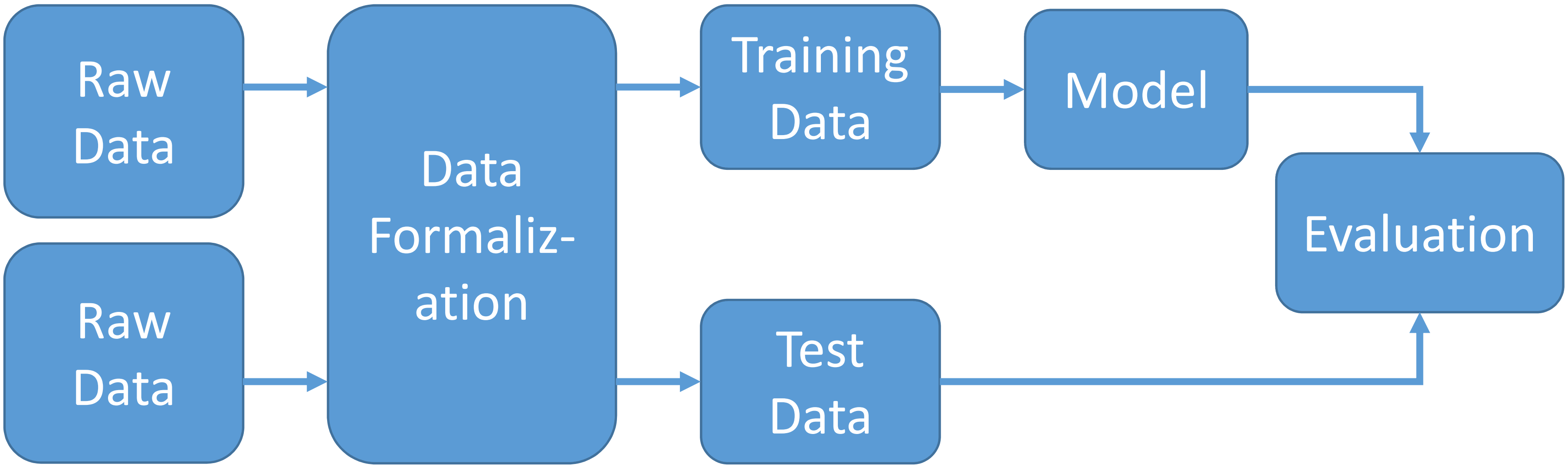
- 2010s
  - Deep learning
  - Learning from big data
  - Learning with GPUs or HPC
  - Multi-task & lifelong learning
  - Deep reinforcement learning
  - Massive applications to vision, speech, text, networks, behavior etc.
  - …

Slide credit: Ray Mooney

Machine Learning Categories

- Supervised Learning
  - To perform the desired output given the data and labels
- Unsupervised Learning
  - To analyze and make use of the underlying data patterns/structures
- Reinforcement Learning
  - To learn a policy of taking actions in a dynamic environment and acquire rewards

Machine Learning Process



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

Supervised Learning

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

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

  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 close to the corresponding label
    - What is the learning objective?
    - How to update the parameters?

Learning Objective

- Make the prediction closed to the corresponding label

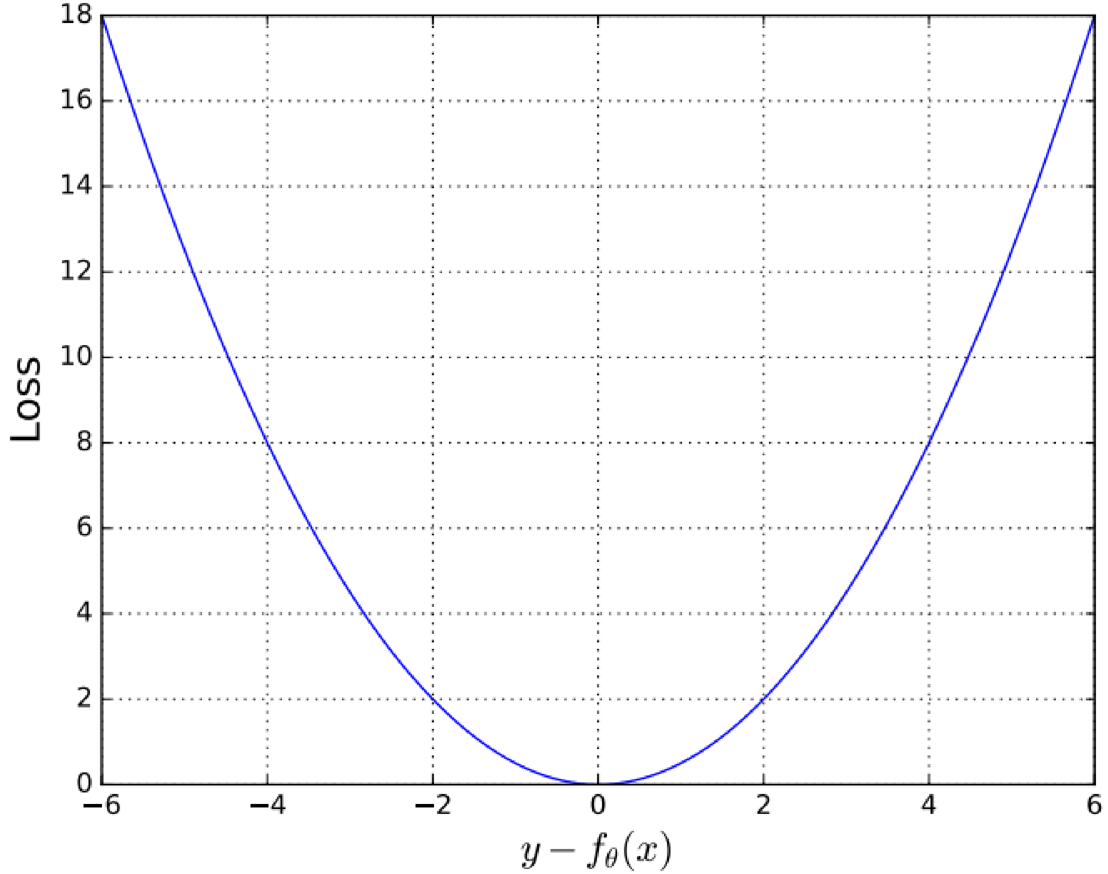
$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_\theta(x_i))$$

- 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

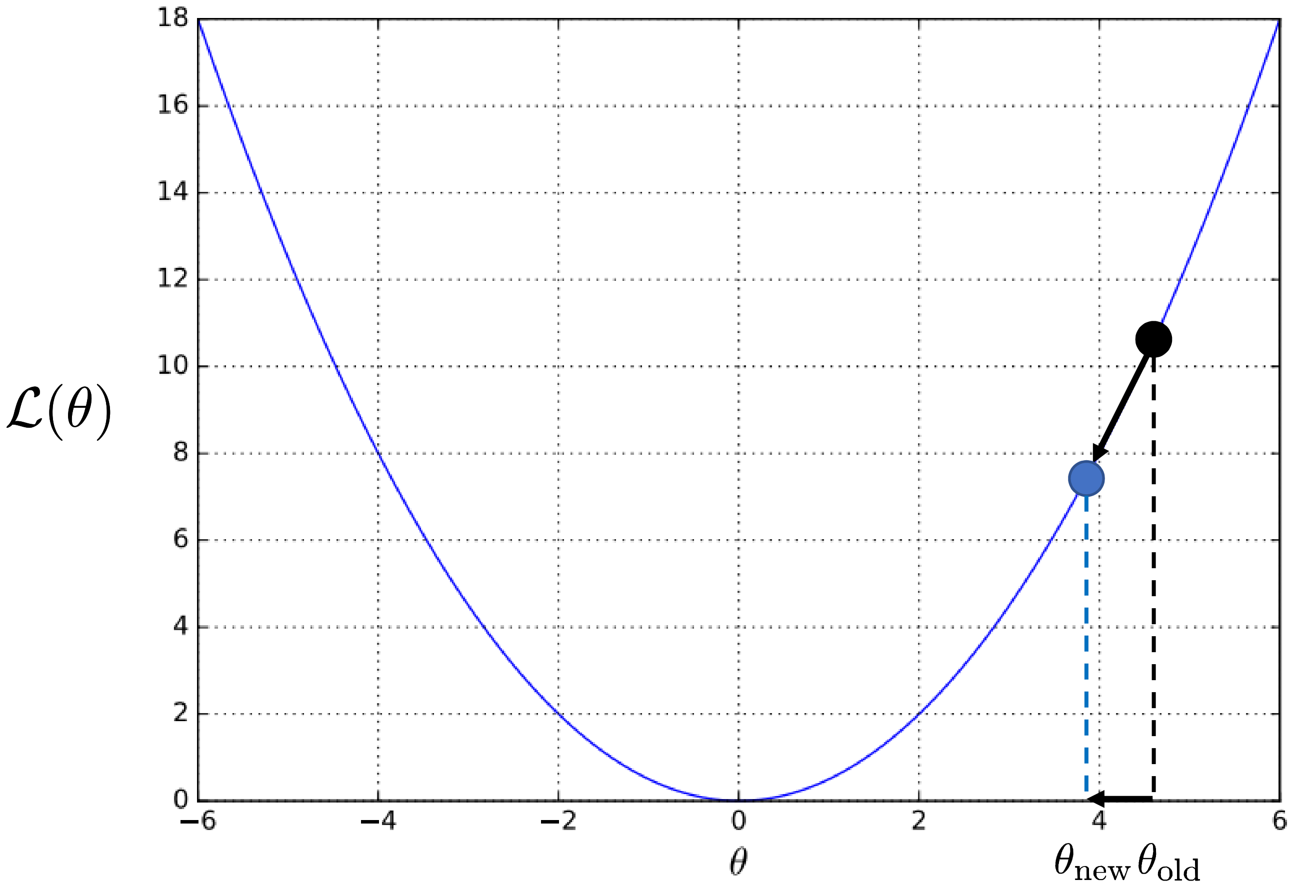
$$\mathcal{L}(y_i, f_\theta(x_i)) = \frac{1}{2}(y_i - f_\theta(x_i))^2$$



- Penalty much more on larger distances

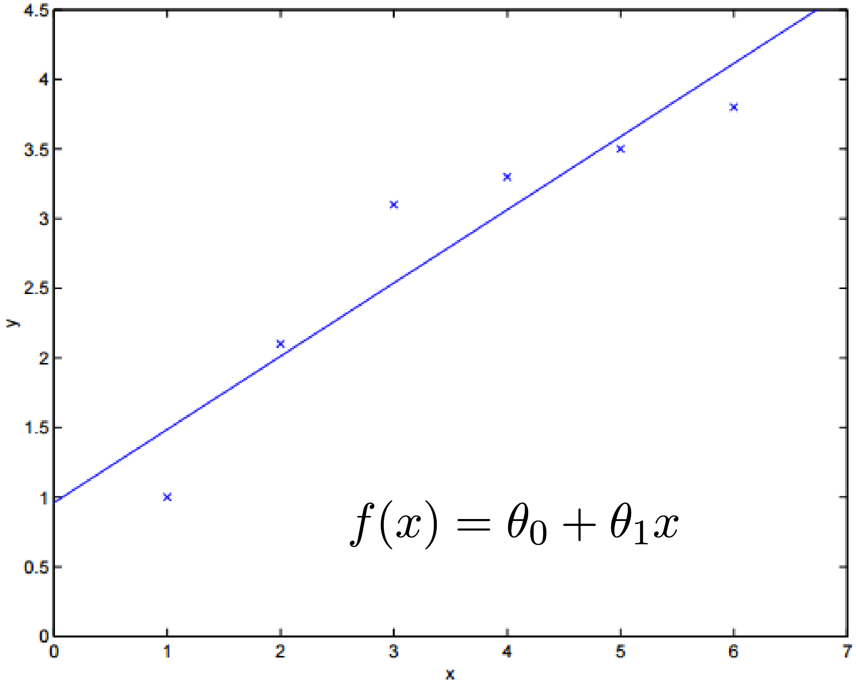
- Accept small distance (error)
  - Observation noise etc.
  - Generalization

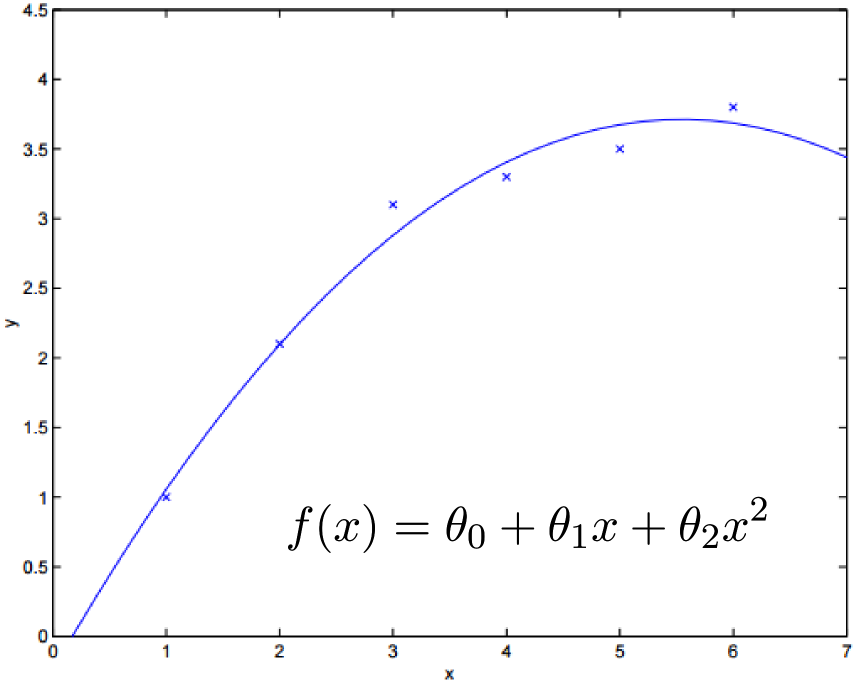
Gradient Learning Methods



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

A Simple Example

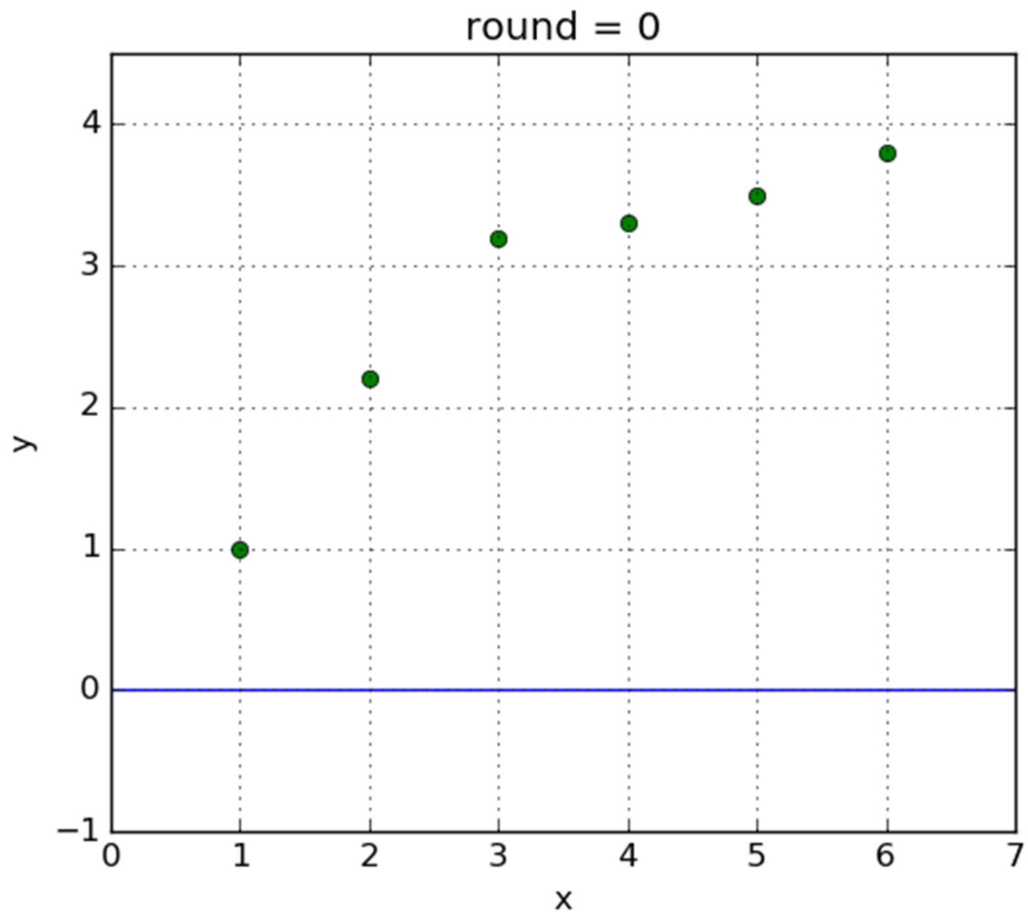




- Observing the data $\{(x_i, y_i)\}_{i=1,2,\ldots,N}$, we can use different models (hypothesis spaces) to learn
  - First, model selection (linear or quadratic)
  - Then, learn the parameters

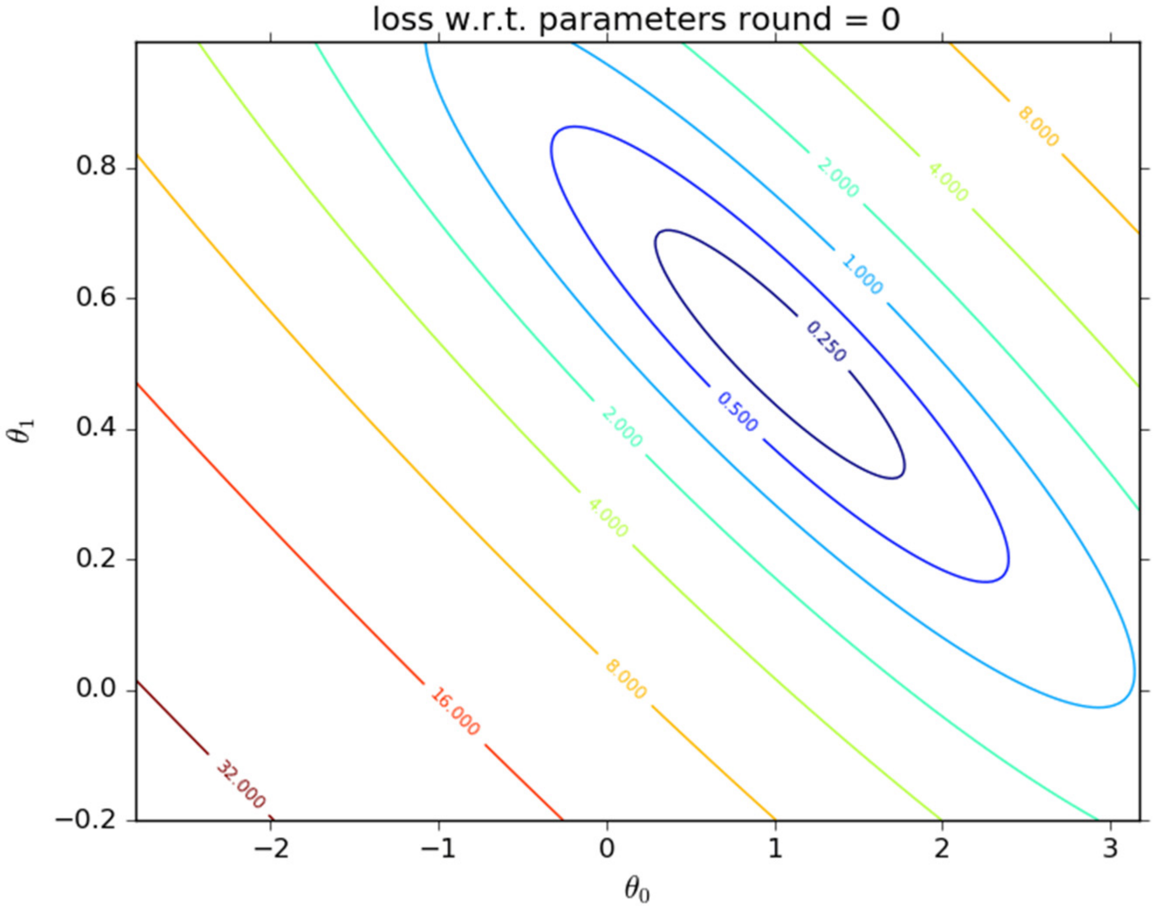
An example from Andrew Ng

Learning Linear Model - Curve



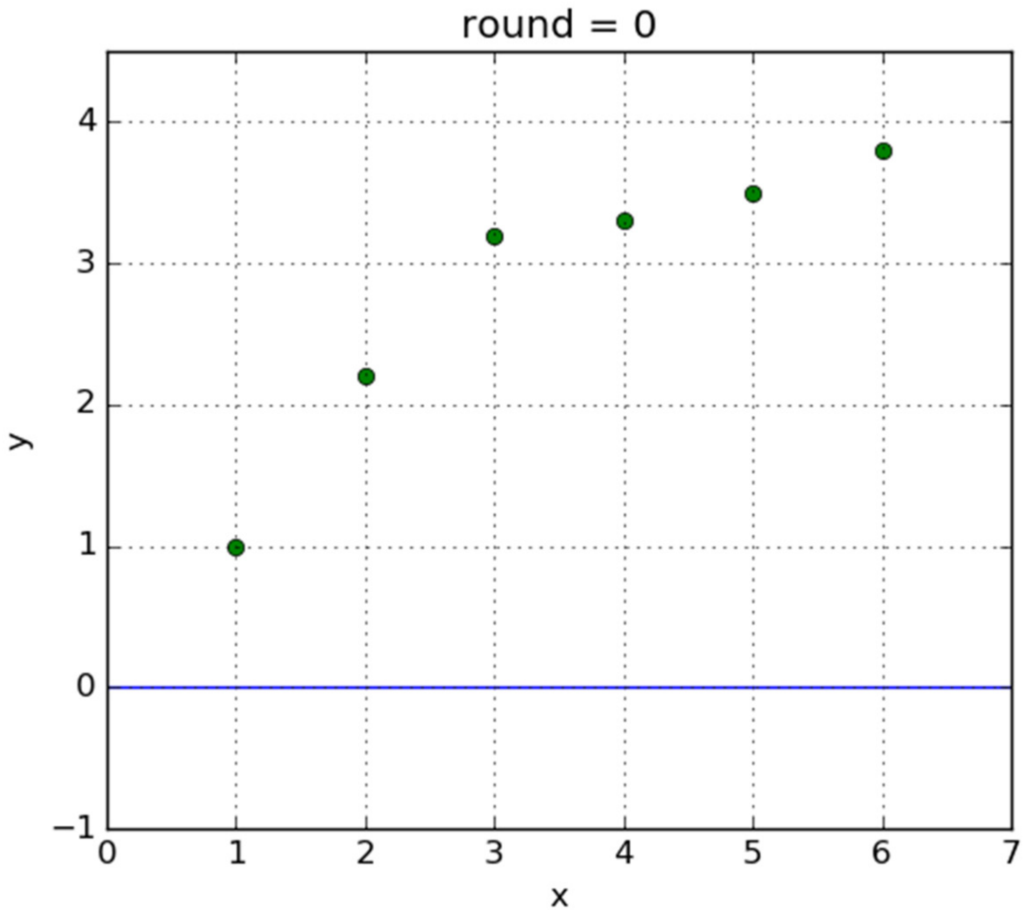
$$f(x) = \theta_0 + \theta_1 x$$

Learning Linear Model - Weights



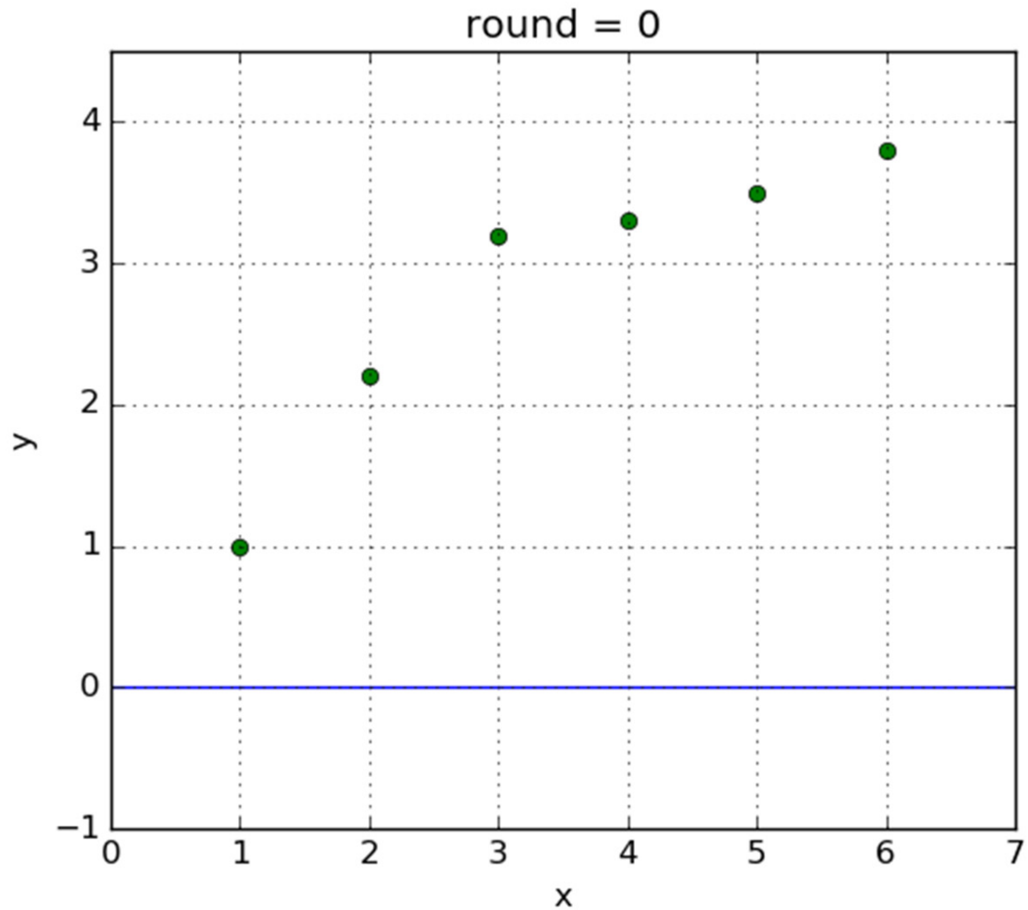
$$f(x) = \theta_0 + \theta_1 x$$

Learning Quadratic Model



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

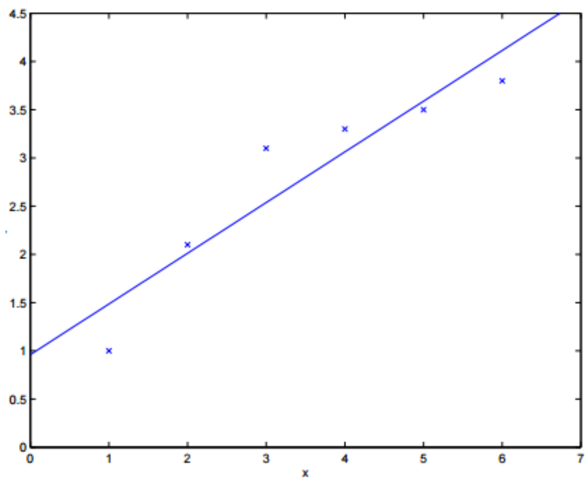
Learning Cubic Model



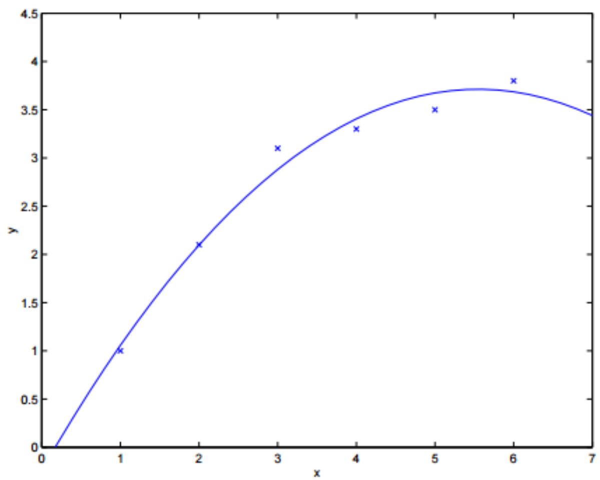
$$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

Model Selection

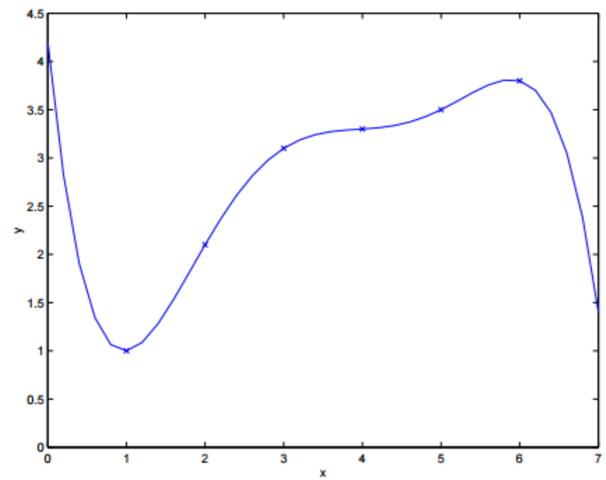
- Which model is the best?



Linear model: underfitting



Quadratic model: well fitting

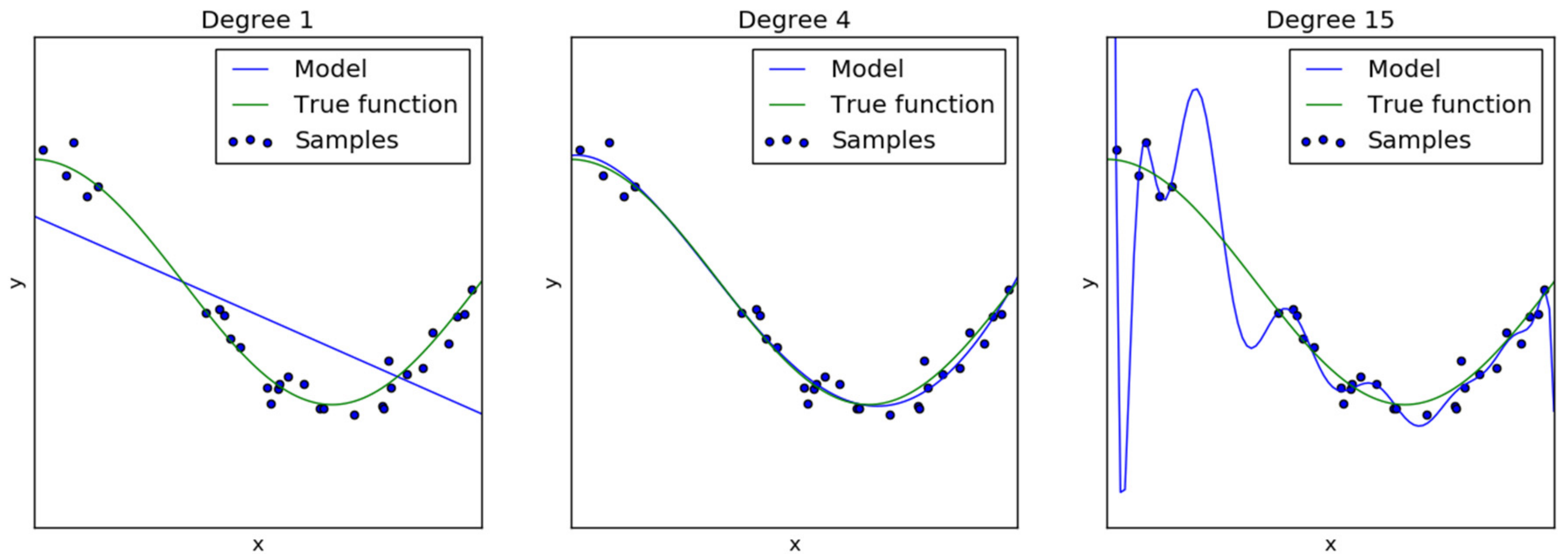


5th-order model: overfitting

- 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

Model Selection

- Which model is the best?



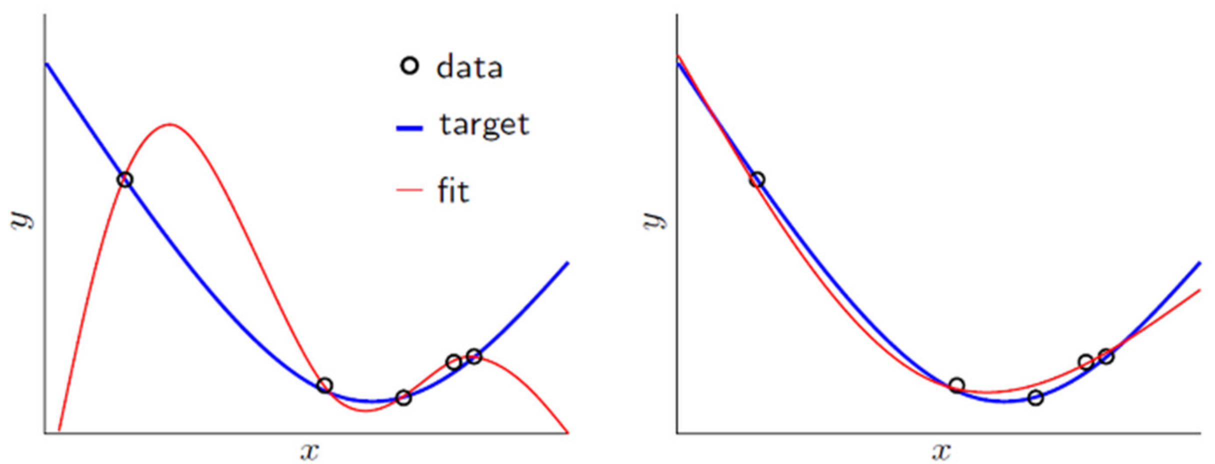
Linear model: underfitting 4^{th}-order model: well fitting 15^{th}-order model: overfitting

- 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

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



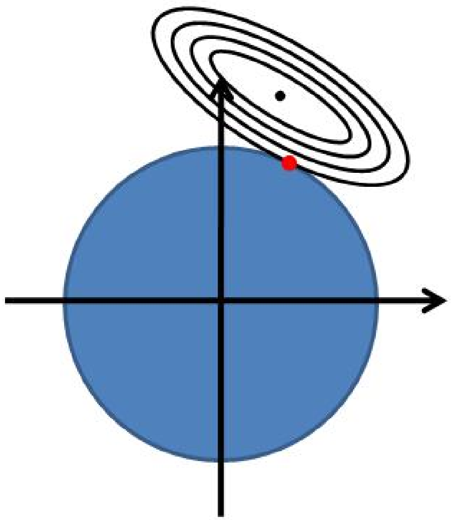
(a) without regularization (b) with regularization

Typical Regularization

- L2-Norm (Ridge)

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

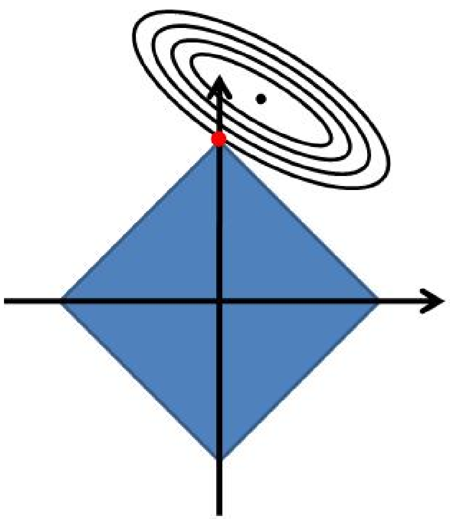
$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_\theta(x_i)) + \lambda ||\theta||_2^2$$



- L1-Norm (LASSO)

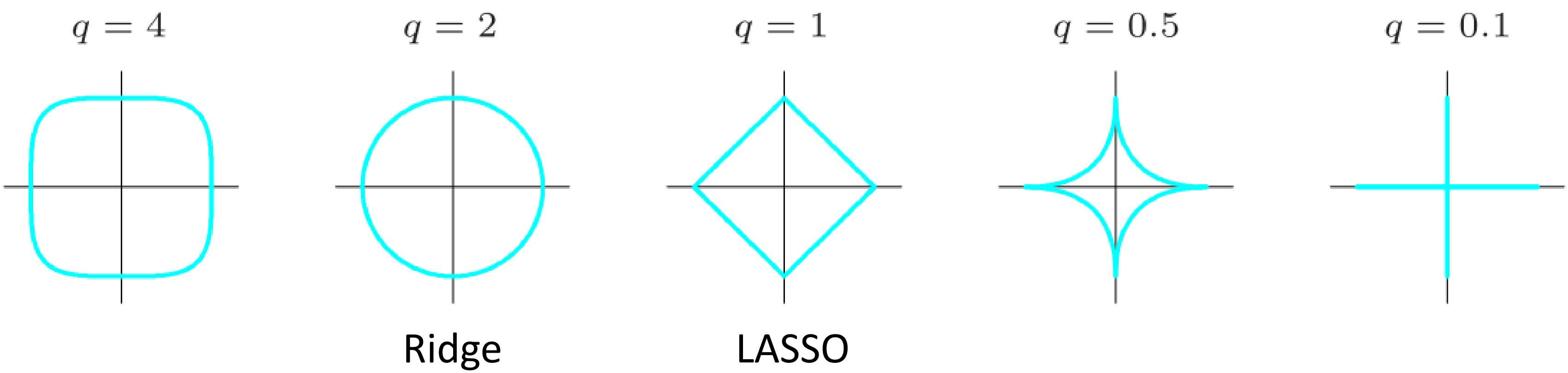
$$\Omega(\theta) = ||\theta||_1 = \sum_{m=1}^{M} |\theta_m|$$

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



More Normal-Form Regularization

- Contours of constant value of $\sum_j |\theta_j|^q$



- Sparse model learning with *q* not higher than 1
- Seldom use of *q* > 2
- Actually, 99% cases use *q* = 1 or 2

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

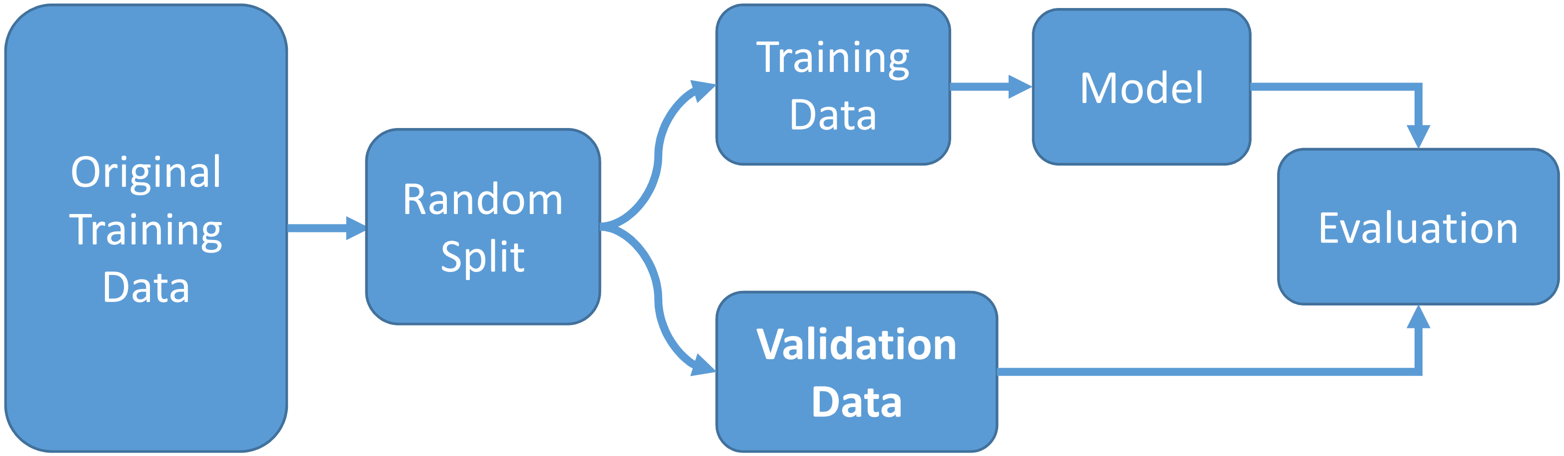
$$\min_\theta \underbrace{\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(y_i, f_\theta(x_i))}_{\text{Original loss}} + \lambda \underbrace{\Omega(\theta)}_{\text{Penalty on assumptions}}$$

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 hyperparameter 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.

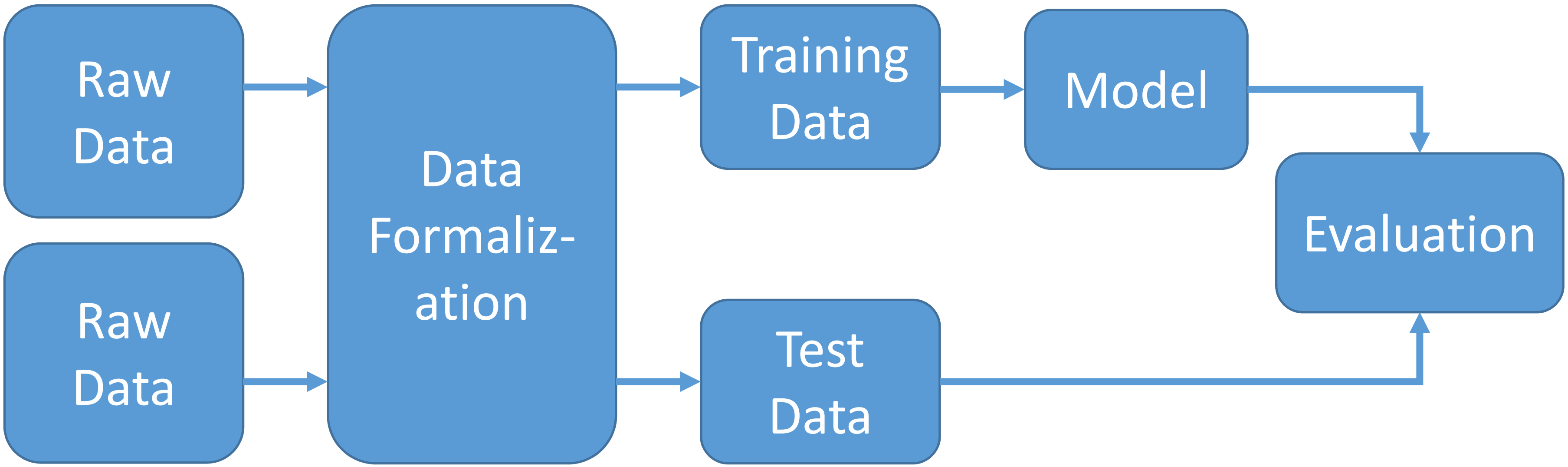
Cross Validation for Model Selection



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

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

A Simple Case Study on Generalization Error

- Finite hypothesis set $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$
- Theorem of generalization error bound:

  For any function $f \in \mathcal{F}$, with probability no less than $1 - \delta$, it satisfies

$$R(f) \leq \hat{R}(f) + \epsilon(d, N, \delta)$$

  where

$$\epsilon(d, N, \delta) = \sqrt{\frac{1}{2N}\left(\log d + \log \frac{1}{\delta}\right)}$$

  - *N*: number of training instances
  - *d:* number of functions in the hypothesis set

Section 1.7 in Dr. Hang Li's text book.

Lemma: Hoeffding Inequality

Let $X_1, X_2, \ldots, X_n$ be bounded independent random variables $X_i \in [a, b]$, the average variable *Z* is

$$Z = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Then the following inequalities satisfy:

$$P(Z - \mathbb{E}[Z] \geq t) \leq \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

$$P(\mathbb{E}[Z] - Z \geq t) \leq \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

http://cs229.stanford.edu/extra-notes/hoeffding.pdf

Proof of Generalized Error Bound

- Assume the bounded loss function $L(y, f(x)) \in [0, 1]$
- Based on Hoeffding Inequality, for $\epsilon > 0$, we have

$$P(R(f) - \hat{R}(f) \geq \epsilon) \leq \exp(-2N\epsilon^2)$$

- As $\mathcal{F} = \{f_1, f_2, \ldots, f_d\}$ is a finite set, it satisfies

$$\begin{aligned} P(\exists f \in \mathcal{F} : R(f) - \hat{R}(f) \geq \epsilon) &= P(\bigcup_{f \in \mathcal{F}} \{R(f) - \hat{R}(f) \geq \epsilon\}) \\ &\leq \sum_{f \in \mathcal{F}} P(R(f) - \hat{R}(f) \geq \epsilon) \\ &\leq d \exp(-2N\epsilon^2) \end{aligned}$$

Proof of Generalized Error Bound

- Equivalence statements

$$P(\exists f \in \mathcal{F} : R(f) - \hat{R}(f) \geq \epsilon) \leq d \exp(-2N\epsilon^2)$$

$$\Updownarrow$$

$$P(\forall f \in \mathcal{F} : R(f) - \hat{R}(f) < \epsilon) \geq 1 - d \exp(-2N\epsilon^2)$$

- Then setting

$$\delta = d \exp(-2N\epsilon^2) \quad \Leftrightarrow \quad \epsilon = \sqrt{\frac{1}{2N} \log \frac{d}{\delta}}$$

The generalized error is bounded with the probability

$$P(R(f) < \hat{R}(f) + \epsilon) \geq 1 - \delta$$

$\square$

Discriminative Model and Generative Model

- Discriminative model
  - modeling the dependence of unobserved variables on observed ones
  - also called conditional models.
  - Deterministic: $y = f_\theta(x)$
  - Probabilistic: $p_\theta(y|x)$
- Generative model
  - modeling the joint probabilistic distribution of data
  - given some hidden parameters or variables

$$p_\theta(x, y)$$

  - then do the conditional inference

$$p_\theta(y|x) = \frac{p_\theta(x,y)}{p_\theta(x)} = \frac{p_\theta(x,y)}{\sum_{y'} p_\theta(x,y')}$$

Discriminative Model and Generative Model

- Discriminative model
  - modeling the dependence of unobserved variables on observed ones
  - also called conditional models.
  - Deterministic: $y = f_\theta(x)$
  - Probabilistic: $p_\theta(y|x)$

  - Directly model the dependence for label prediction
  - Easy to define dependence-specific features and models
  - Practically yielding higher prediction performance

  - Linear regression, logistic regression, k nearest neighbor, SVMs, (multi-layer) perceptrons, decision trees, random forest etc.

Discriminative Model and Generative Model

- Generative model
  - modeling the joint probabilistic distribution of data
  - given some hidden parameters or variables

$$p_\theta(x, y)$$

  - then do the conditional inference

$$p_\theta(y|x) = \frac{p_\theta(x, y)}{p_\theta(x)} = \frac{p_\theta(x, y)}{\sum_{y'} p_\theta(x, y')}$$

  - Recover the data distribution [essence of data science]
  - Benefit from hidden variables modeling
  - Naive Bayes, Hidden Markov Model, Mixture Gaussian, Markov Random Fields, Latent Dirichlet Allocation etc.