MACHINE LEARNING

CHAPTER 0: INTRODUCTION

Contact Information

Instructor: Qi Hao

E-mail: hao.q@sustc.edu.cn

Office: South Tower Room 415

Office Hours: M 2:00-4:00pm

Available other times by appointment or the open door policy

Phone: 186-6495-7027

QQ: 820786590 2022机器学习

Web: *http://hqlab.isus.tech/teaching/CS405*

BB: Machine Learning Fall 2022

OJ: *http://oj.isus.tech/*

Class Schedule

- **Lectures**: T 8:00 am – 9:50 am Teaching Building I room 506
- **Study**: F 7:00 pm – 8:50 pm Teaching Building I Room 506
- **Grading policy:**

Final Exam (in-class):	**20%**	**Midterm Exam (take-home):**	**10%**
Assignments (8~12 times):	**20%**	**Quizzes (<=10 times):**	**10%**
Lab Projects :	**20%**	**Final Projects (4 per group):**	**20%**
Bonus Credits:	**<=5%**		

90~93:	A-	94~97:	A	98~100:	A+
80~82:	B-	83~86:	B	87~89:	B+
70~72:	C-	73~76:	C	77~79:	C+
60~62:	D-	63~66:	D	67~69:	D+

Textbook and Lecture Notes

Textbooks:

[1] Pattern Recognition and Machine Learning, by Christopher M. Bishop, 2006 Springer

[2] Machine Learning in Action, by Peter Harrington, 2012, Manning

Other books:

[1] 机器学习, 周志华

[2] Dive in Deep Learning, by Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola

[3] Reinforcement Learning: An Introduction, by Richard S. Sutton

[4] The Elements of Statistical Learning, by Trevor Hstie, Rober Tibshirani, Jerome Friedman

Paper reading:

[1] Ghahramani Z. Probabilistic machine learning and artificial intelligence, Nature, 2015

[2] Lecun Y, Bengio Y, Hinton G. Deep learning, Nature, 2015

[3] Littman M L. Reinforcement learning improves behavior from evaluative feedback, Nature, 2015

Lecture notes:

http://hqlab.isus.tech/teaching/CS405

Other Resources

Assignment platform: bb.sustech.edu.cn

Textbook resource: https://www.microsoft.com/en-us/research/people/cmbishop/#prml-book

Textbook Matlab codes : http://prml.github.io/

Matlab toolboxes : Machine Learning, Neural Networks

Teaching Objectives

- Fundamental knowledge about machine learning and pattern recognition, from Bayesian approaches to deep learning frameworks through lectures, quizzes and exercises

- Machine learning system development methods in Python based platforms (numpy, sciki-learn, pytorch) through labs and projects

- Model-based and data-driven machine learning system design and integration skills through the final project, literature surveys and reports

Lecture Schedule

Section 0	Course Introduction	
Section 1	Preliminary	(HW1)
Section 2	Probability Distributions	(HW2)
Section 3	Linear Regression and Classification	(HW3)
Section 4*	Dimension Reduction and Feature Selection	(HW4)
Section 5	Neural Networks	(HW5)
Section 6	Sparse Kernel Machine	(HW6)
Section 7	Clustering and EM learning	(HW7)
-- *Midterm Exam* --		
Section 8*	Ensemble Learning	(HW8)
Section 9	Hidden Markov Models	(HW9)
Section 10*	Bayesian Networks	(HW10)
Section 11	Markov Decision Process	(HW11)
Section 12*	Reinforcement Learning	(HW12)
-- *Final Exam* --		

Lab Schedule

Section 0	Lab Introduction
Section 1	Preliminary
Section 2	Bayes
Section 3	Regression and Classification
--Final Project Proposal--	
Section 4	Decision Tree
Section 5	Random Forest (Ensemble Learning)
Section 6	KNN and Support Vector Machine
Section 7	K-Mean and EM Clustering
Section 8	Neural Network (I)
Section 9	Neural Network (II)
Section 10	Neural Network (III)
Section 11	Reinforcement Learning
--Final Project Report--	

Final Project Examples

[1] Reinforcement learning based planning using a self-driving car simulator

[2] Segmentation of 2D/3D measurements for self-driving applications

[3] Detection and recognition of traffic signs for self-driving applications

[4] Detection and tracking of 2D/3D objects for self-driving applications

[5] Federated learning for model fusion of networked vehicle applications

[6] GNN for self-driving data augmentation

Bonus Credits

- AI companies
- Survey Papers
- Attendance
- Bonus Credits

Plagiarism

From Spring 2018, the plagiarism policy applied by the Computer Science and Engineering department is the following:

* If an assignment is found to be plagiarized, the first time the score of the assignment will be 0.

- The second time the score of <u>the course</u> will be 0.

As **it may be difficult** when two assignments are identical or nearly identical who actually wrote it, the policy will apply to BOTH students, unless one confesses having copied without the knowledge of the other.

What is OK, and what isn't OK

It's OK

- to work on an assignment with a friend, and think together about the program structure, share ideas and even the global logic. At the time of actually writing the code, you should write it alone.
- to use in an assignment a piece of code found on the web, as long as you indicate in a comment where it was found and don't claim it as your own work.
- to help friends debug their programs (you'll probably learn a lot yourself by doing so).
- to show your code to friends to explain the logic, as long as the friends write their code on their own later.

It's NOT OK

- **to take the code of a friend, make a few cosmetic changes (comments, some variable names) and pass it as your own work.**

Make a Promise to Keep

Sign

the “Student Commitment for Assignments”

Keep

the promise during the whole semester!

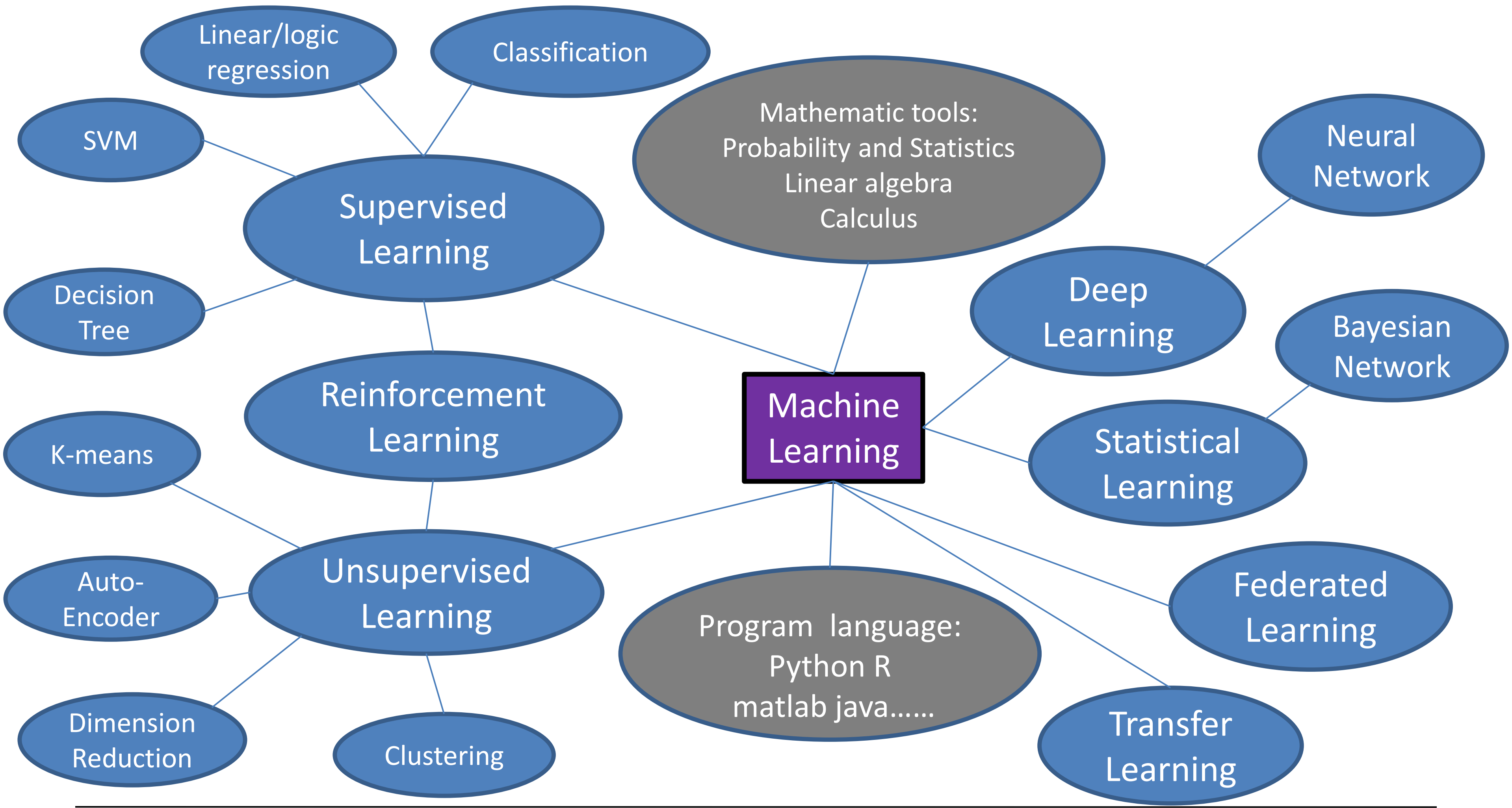
Learning Objectives

1. What is the history of machine learning?
2. What are the most important functionalities of machine learning?
3. What are the major technical challenges for developing machine learning systems?
4. What are the most useful tools for developing machine learning systems?
5. What are the most popular software and hardware platforms for developing machine learning systems?
6. What are the most promising applications for machine learning?

Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Framework



Linear/logic regression
Classification
SVM
Supervised Learning
Mathematic tools:
Probability and Statistics
Linear algebra
Calculus
Neural Network
Decision Tree
Deep Learning
Bayesian Network
Reinforcement Learning
Machine Learning
Statistical Learning
K-means
Unsupervised Learning
Auto-Encoder
Program language:
Python R
matlab java……
Federated Learning
Dimension Reduction
Clustering
Transfer Learning

Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Problem Statement

- **Problem**：Predict the label $\hat{Y}$ and data $\hat{X}$ with training set（X,Y）？

Unknown model

Training label　　　Training data

Training： $Y = \underline{A} \circ X$

Prediction： $\hat{Y} = A \circ \hat{X}$

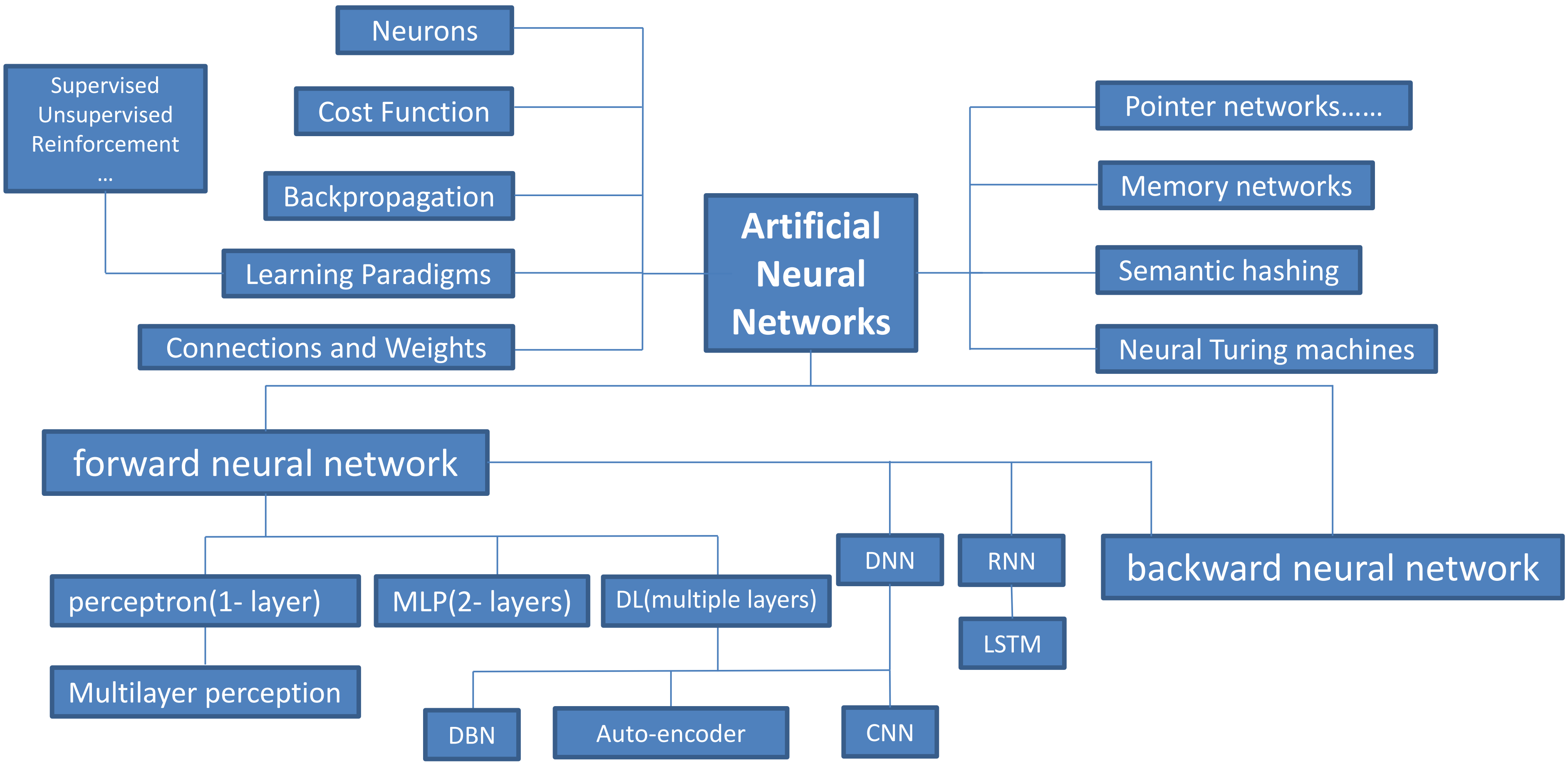
Predicting label　　　Predicting data

$$\begin{cases} Y\ and\ X\ is\ known\text{： supervised learning} \\ Y\ or\ X\ is\ unknown\text{： unsupervised learning} \end{cases}$$

$$\begin{cases} Y\text{、}\hat{Y}\ are\ continuous\text{： Regression} \\ Y\text{、}\hat{Y}\ are\ discrete\text{： classification} \end{cases}$$

$Y\ is\ known\ and\ Dim(Y) > Dim(X)$： dimensionality reduction

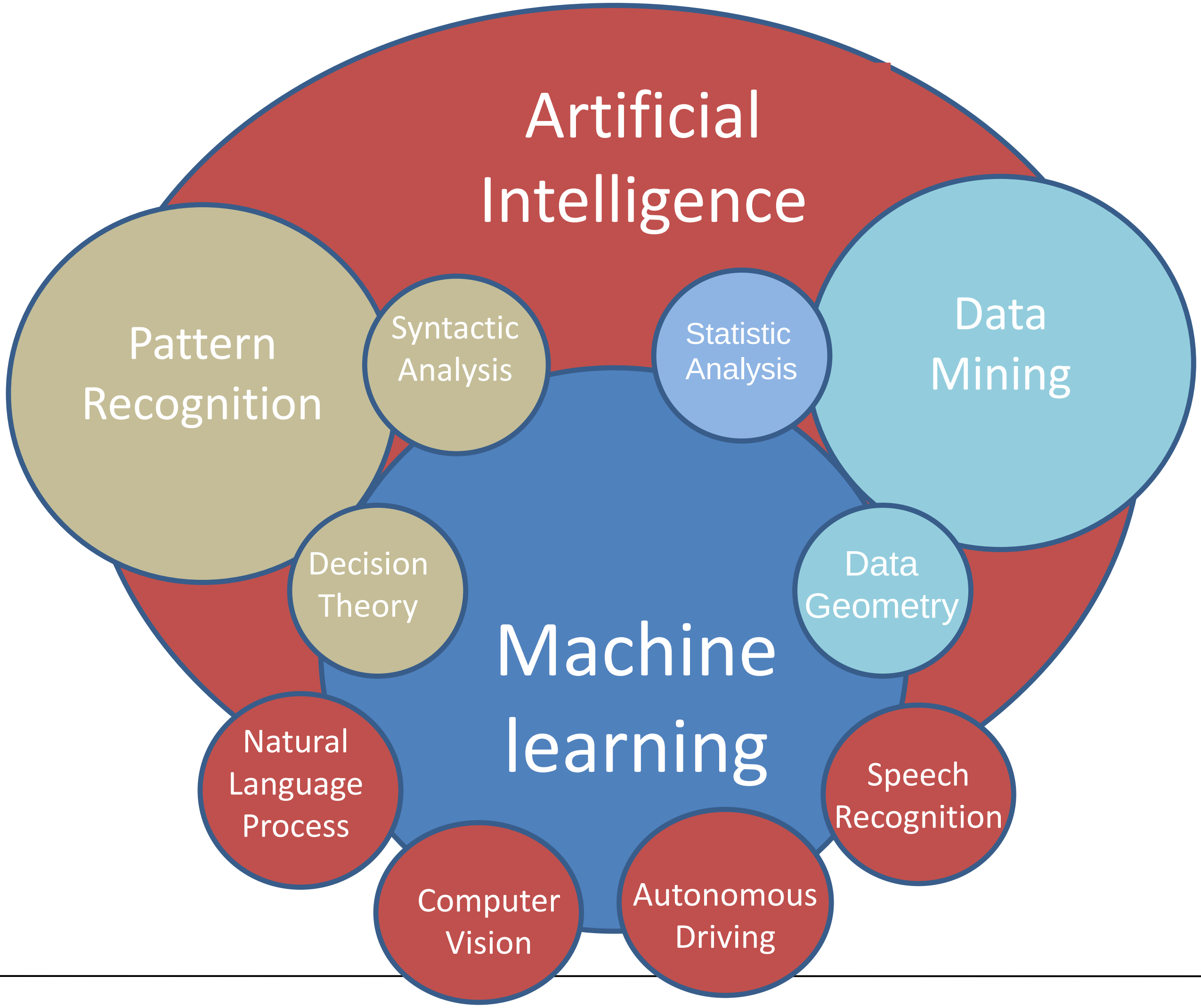
Neural Network Models



Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

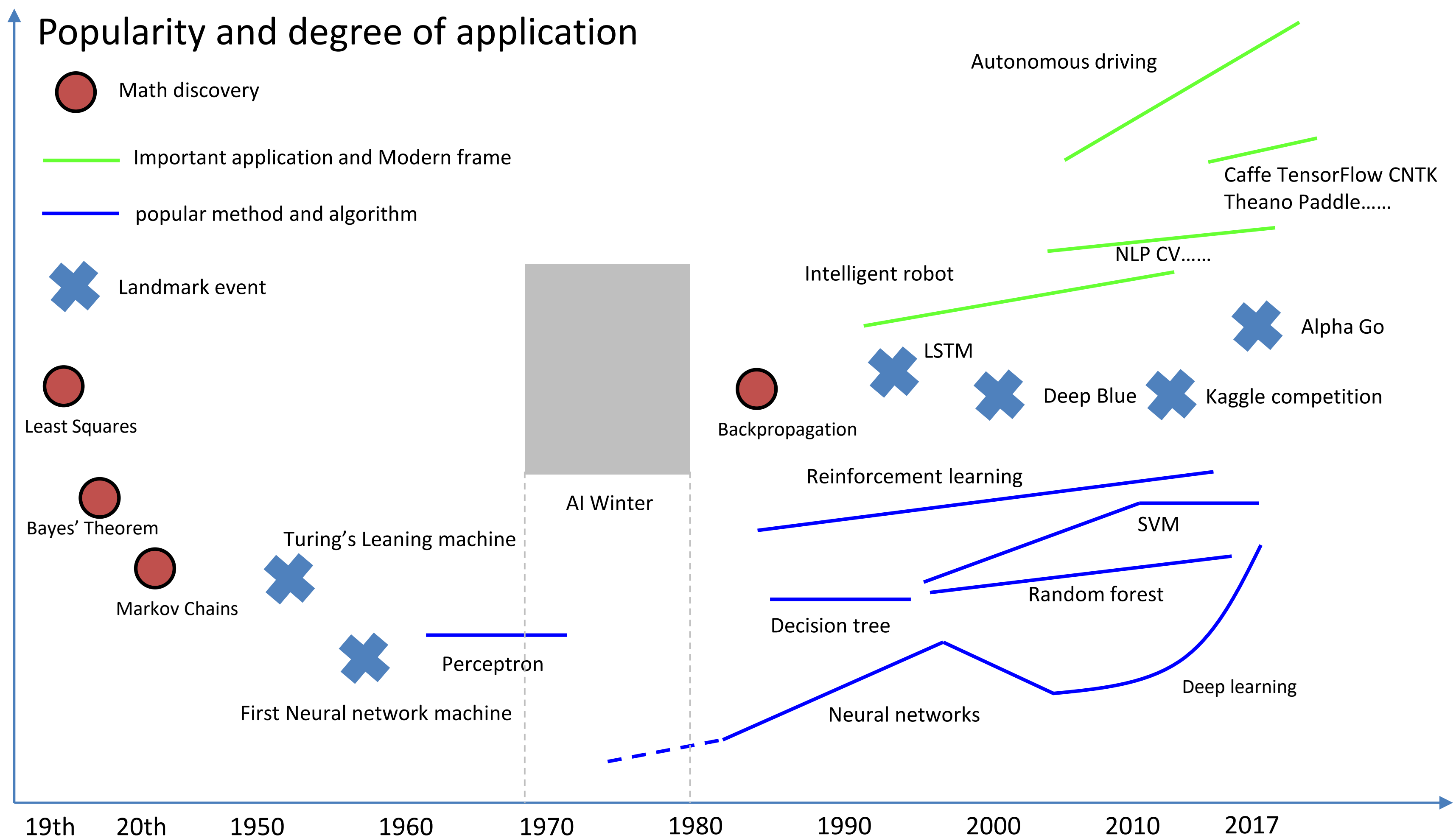
The Whole Picture



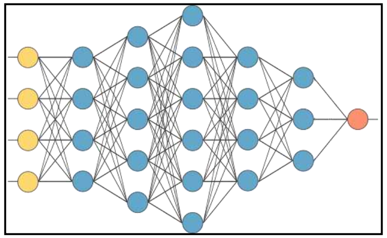
Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

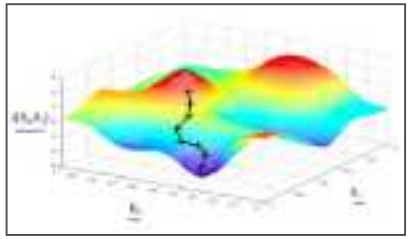
History


Popularity and degree of application
Math discovery
Important application and Modern frame
popular method and algorithm
Landmark event
Least Squares
Bayes' Theorem
Markov Chains
Turing's Leaning machine
First Neural network machine
Perceptron
AI Winter
Backpropagation
Intelligent robot
LSTM
Deep Blue
Kaggle competition
Alpha Go
Autonomous driving
Caffe TensorFlow CNTK
Theano Paddle……
NLP CV……
Reinforcement learning
SVM
Random forest
Decision tree
Neural networks
Deep learning
19th
20th
1950
1960
1970
1980
1990
2000
2010
2017

Deep ML vs Conventional ML



model



optimization

data

platform

CNN

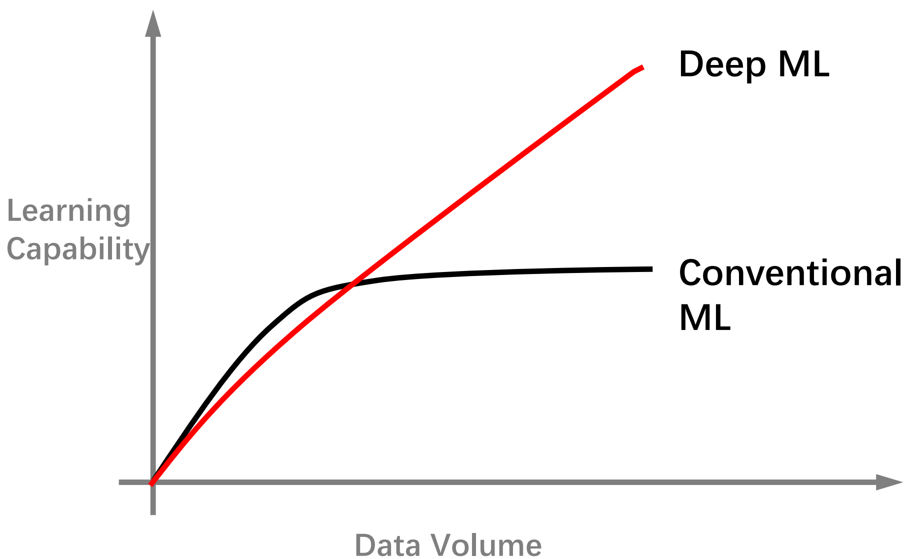
RNN

LSTM

Auto Encoder

GAN

RL & FL



GPU & TPU

Cloud Computing

Data Visualization

GPU Service

TensorFlow

Caffe & Pytorch

Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Machine learning

- **Machine Learning—minimization of some loss function for generalizing data sets with models.**

- **Datasets —annotated, indexed, organized**

- **Models —tree, distance, probabilistic, graph, bio-inspired**

- **Optimization —algorithms can minimize the loss.**

Datasets

- Collection
- Storage
- Annotation
- Indexing
- Organization
- Access

Simulators

- Data visualization
- Generate training data
- Algorithm evaluation

Benchmark Metrics

- System functionalities
- System scalability
- System robustness
- System efficiency

Models

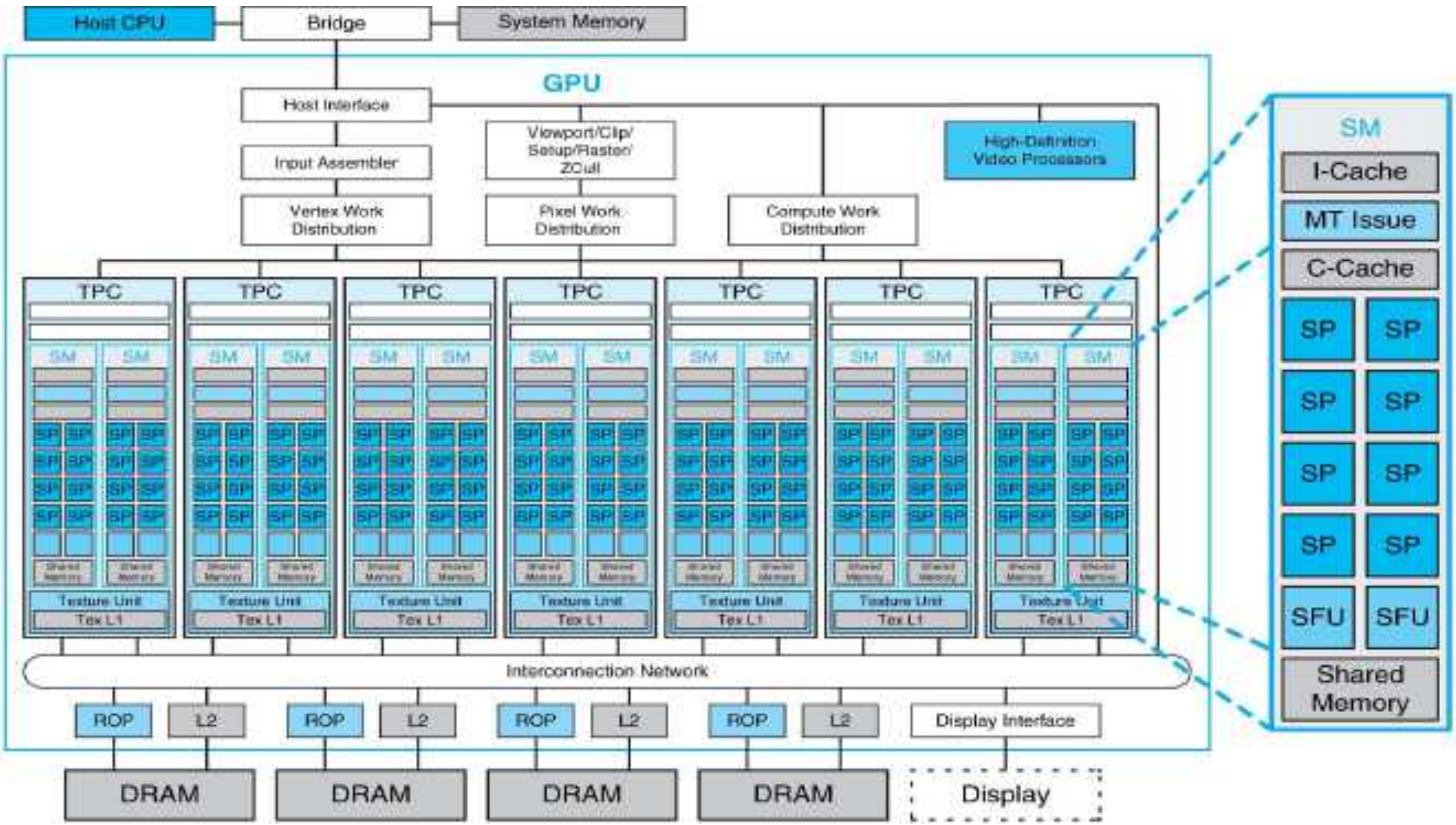
- Tree Models
- Distance-based Models
- Probabilistic Models
- Neural Network Models
- Graph-based Models

Models

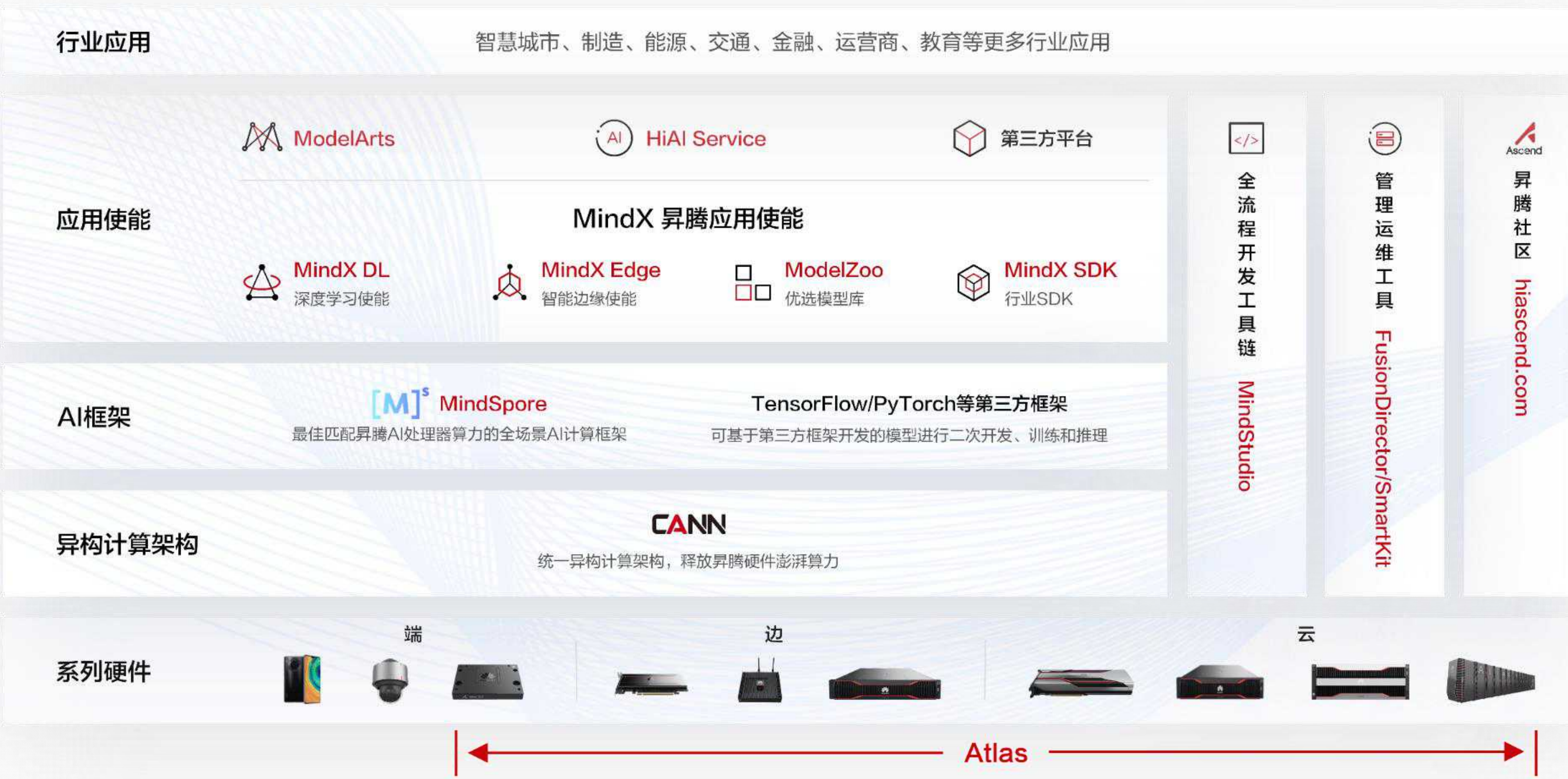
- Boosting
- Mixed Models
- Ensemble Learning

Hardware Platform (GPU Server)

Hardware Platform (GPU)



构筑业界最强AI算力平台，极简易用、极致性能



Atlas系列硬件打造人工智能算力平台基石

Atlas训练系列硬件

Atlas 300T训练卡

单卡算力**业界领先**

320 TFLOPS FP16

Atlas 800

训练服务器

Atlas 900 PoD

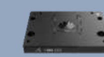
Atlas 900 AI集群

Atlas推理系列硬件

Atlas 300I 推理卡

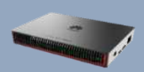
Atlas 800

推理服务器



Atlas 200

AI加速模块



Atlas 200DK

Atlas 500

智能小站

Atlas 500 Pro

智能边缘服务器

异构计算架构CANN，软硬协同充分释放澎湃算力

使能全场景

- 支持**10+**端边云设备形态
- 支持**14+**操作系统
- 支持多种AI框架

使能极简开发

- 统一API，AscendCL
- 自定义算子开发方式
  TBE-DSL/TBE-TIK/AI CPU

使能极致性能

- 亲和昇腾的图编译技术
- **1200+**高性能算子

开源AI框架MindSpore，构建端边云全场景生态

简单的开发体验

帮助开发者实现网络自动切分，只需串行表达就能实现并行训练，降低门槛，简化开发流程。

灵活的调试模式

具备训练过程静态执行和动态调试能力，开发者通过变更一行代码即可切换模式，快速在线定位问题。

充分发挥硬件潜能

最佳匹配昇腾处理器，最大程度地发挥硬件能力，帮助开发者缩短训练时间，提升推理性能。

全场景快速部署

支持云、边缘和手机上的快速部署，实现更好的资源利用和隐私保护，让开发者专注于AI应用的创造。

全自动并行

静态图自动混合并行
训练**性能提升40%**

动态图优化
性能**超越业界60%**

全场景协同

云端分布式推理

边缘AI加速

超轻量IoT设备推理

全流程极简

第三方框架转换工具
业务快速迁移

开发者生态

51万+
下载量

2300+
社区贡献者

开放AI应用使能套件MindX，加速人工智能应用创新

MindX：昇腾应用使能

MindX DL	MindX Edge	**MindX SDK**	ModelZoo
深度学习使能	智能边缘使能	行业应用开发套件	**250+**预训练模型

MindX SDK：沉淀行业知识，使能行业应用 极简开发

2人月
传统应用开发方式

2人天
基于SDK开发方式

已支撑 **20+** 场景化解决方案高效开发

华为松山湖产线		友达光电	南瑞继远		...
PCB板质检	*散热器缺失检测*	*切片AOI检测*	*变电站火警检测*	*变电站人员着装检测*	

一站式开发环境MindStudio，打造高效、便捷的全流程开发工具链

MindStudio是一套基于IntelliJ框架的开发工具平台。提供了应用开发、调试、模型转换功能，同时还提供了网络移植、优化和分析等功能，为用户开发应用程序带来了极大的便利。

- 训练脚本转换
- 模型转换
- 精度比对
- Profiling性能分析
- System Profiling工具
- AI Core Error分析工具

应用开发

- 基于MindX SDK开发应用
- 基于离线工程开发应用
- 应用工程调试

模型开发

- 查询模型
- 模型训练
- 模型可视化

算子开发

- 查询算子
- 开发流程
- 算子分析
- 工程创建
- TBE算子开发（TensorFlow）
- TBE算子开发（PyTorch）
- TBE算子开发（MindSpore）
- AI CPU算子开发（TensorFlow）

Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Machine learning and Optimization

- **Machine Learning—minimization of some loss function for generalizing data sets with models.**
- **Datasets —annotated, indexed, organized**
- **Models —tree, distance, probabilistic, graph, bio-inspired**
- **Optimization —algorithms can minimize the loss.**

What is optimization?

- Finding (one or more) minimizer of a function subject to constraints

$$arg \min_{x} f_0(x)$$

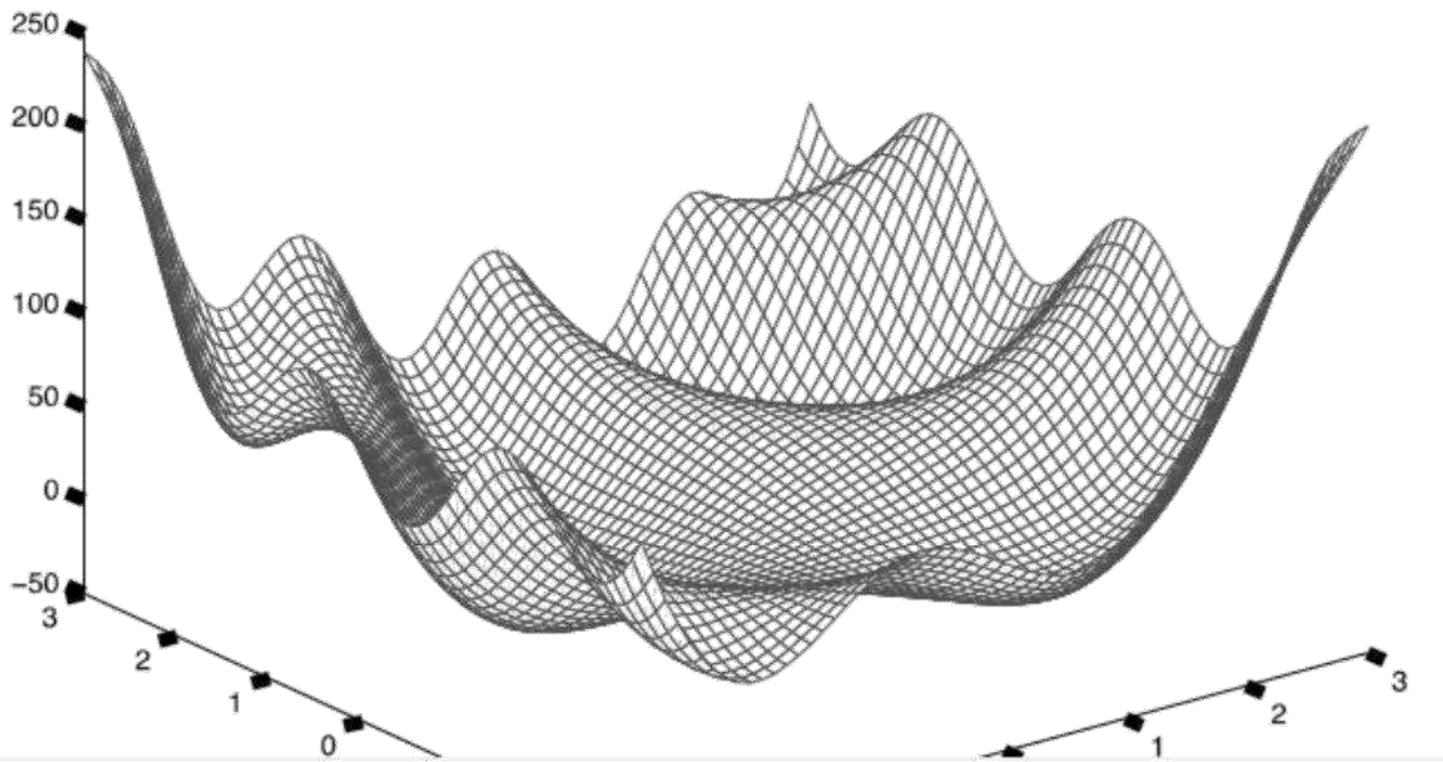
$$s.t. f_i(x) \leq 0, i = \{1, \dots, k\}$$

$$s.t. h_j(x) = 0, j = \{1, \dots, l\}$$

- Most of the machine learning problems are, in the end, optimization problems

General Problem

- Minimize f(x)



Linear Optimization

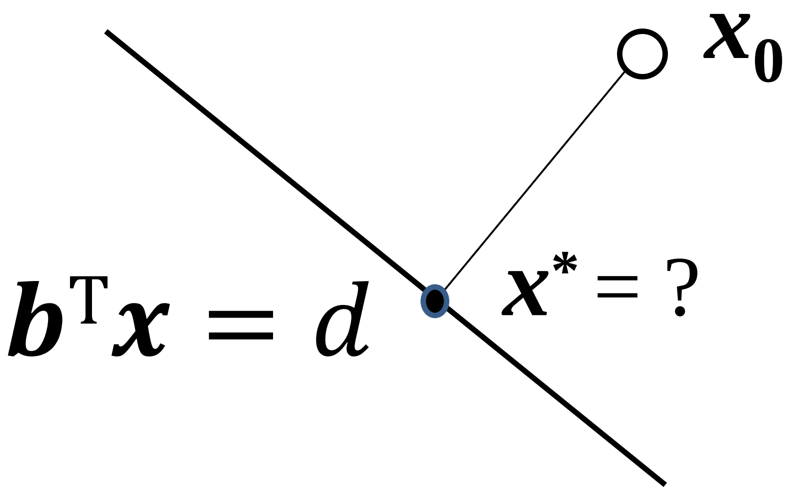
$$\boxed{Y = AX + w} \qquad w \sim \mathcal{N}(0, R)$$

$$X^* = \min_X (Y - AX)^{\mathrm{T}} R^{-1} (Y - AX)$$

$$\frac{\partial}{\partial X^T} (Y - AX)^{\mathrm{T}} R^{-1} (Y - AX) = 0$$

$$\Rightarrow X^* = (A^T R^{-1} A)^{-1} A^T R^{-1} Y$$

Linear Optimization



$$\boxed{\boldsymbol{x}^* = \boldsymbol{x}_0 - \frac{(\boldsymbol{b}^T\boldsymbol{x}_0 - d)\boldsymbol{b}}{\boldsymbol{b}^T\boldsymbol{b}}}$$

$$\boldsymbol{x}^* = \min_{\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{x}_0)^T(\boldsymbol{x} - \boldsymbol{x}_0)$$

$$\text{s.t. } \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x} - d = 0$$

Nonlinear Optimization

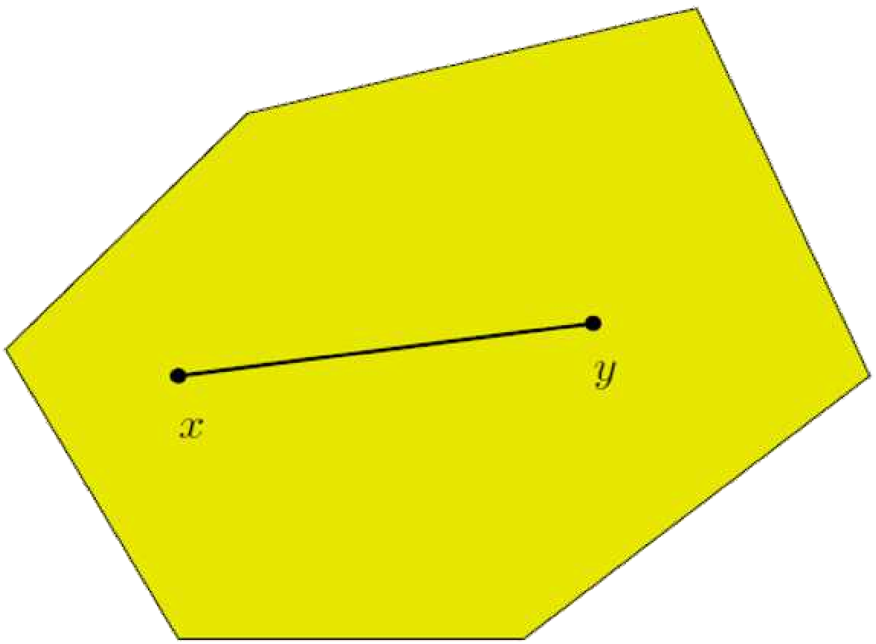
- Convex Optimization
  - Unconstrained optimization
  - Constrained optimization
  - SVMs and Bayesian models
- Non-convex Optimization
  - Heuristic algorithms
  - Random search

What is Convex?

- Convex sets

Def: A set $C \subseteq R$ is convex if for $x,y \in C$; $a \in [0,1]$

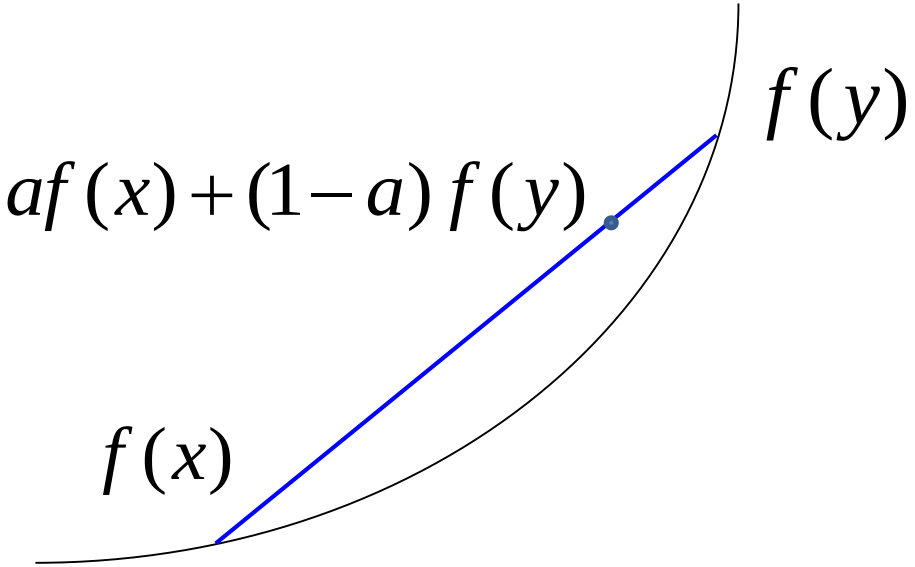
$$ax + (1-a)y \in C$$



What is Convex?

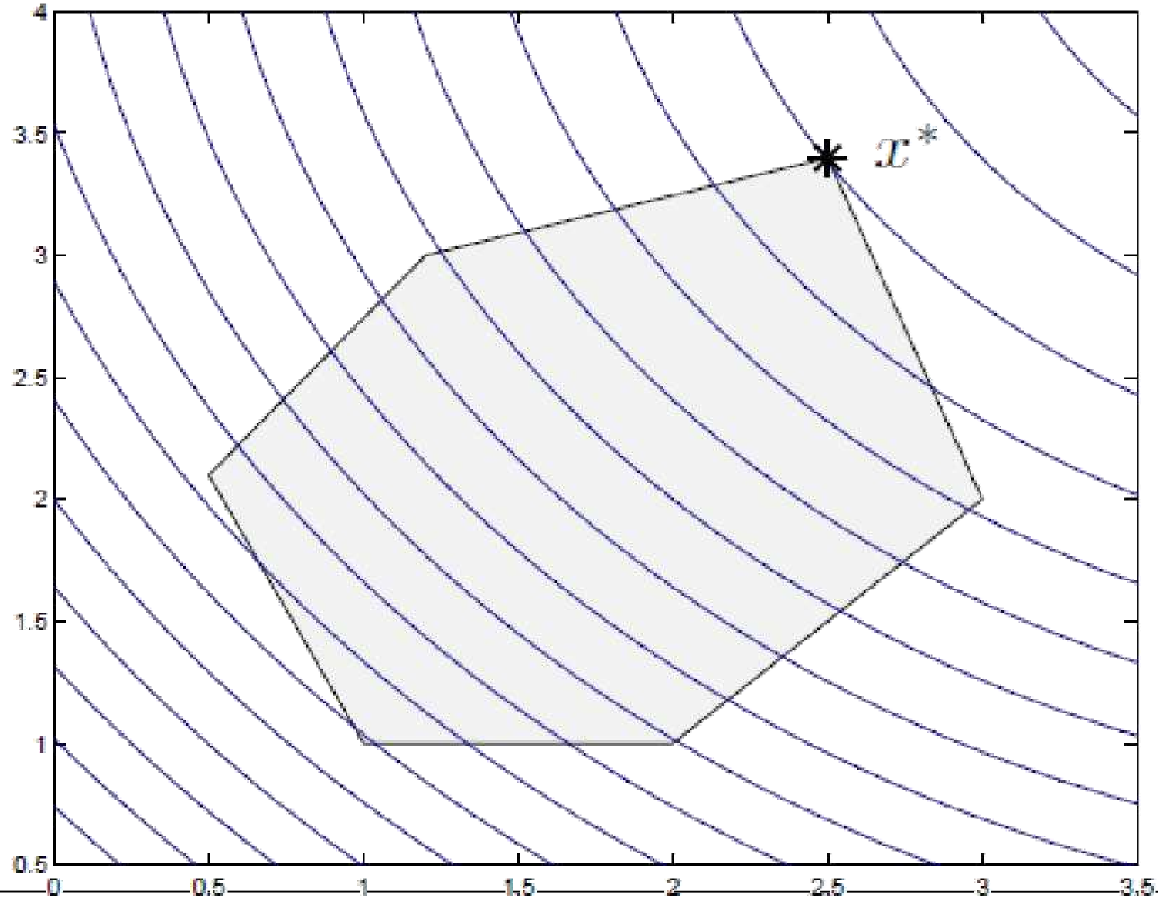
- Convex functions

$$f(ax+(1-a)y) \le af(x)+(1-a)f(y)$$



Convex Optimization

- Local minimizer = Global minimizer



Convex Optimization

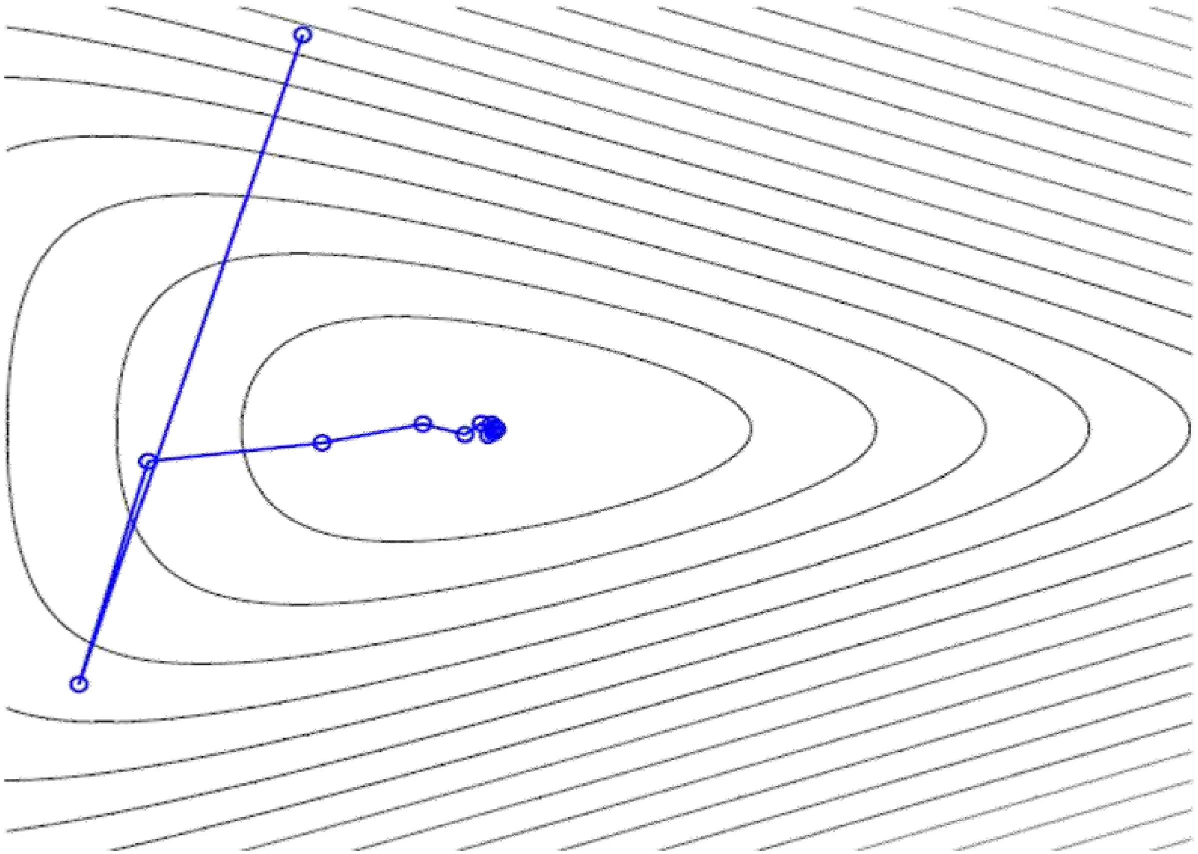
- Unconstrained optimization
  - Gradient descent
  - Gauss-Newton’s method
  - Batch learning
  - Stochastic Gradient Descent
- Constrained optimization
  - Lagrange methods
  - Bayesian methods

Convex optimization

- Unconstrained optimization
  - Gradient descent
  - Gauss-Newton's method
  - Batch learning
  - Stochastic Gradient Descent
- Constrained optimization
  - Lagrange methods
  - Bayesian methods

Gradient Descent

$$f(x_{t+1}) = f(x_t) - \eta \nabla f(x_t)^T (x - x_t)$$



Gauss-Newton's Method

- Idea: use a second-order approximation to function

$$f(x+\Delta x) \approx f(x) + \nabla f(x)^T \Delta x + \frac{1}{2}\Delta x^T \nabla^2 f(x) \Delta x$$

- Choose Δx to minimize above:

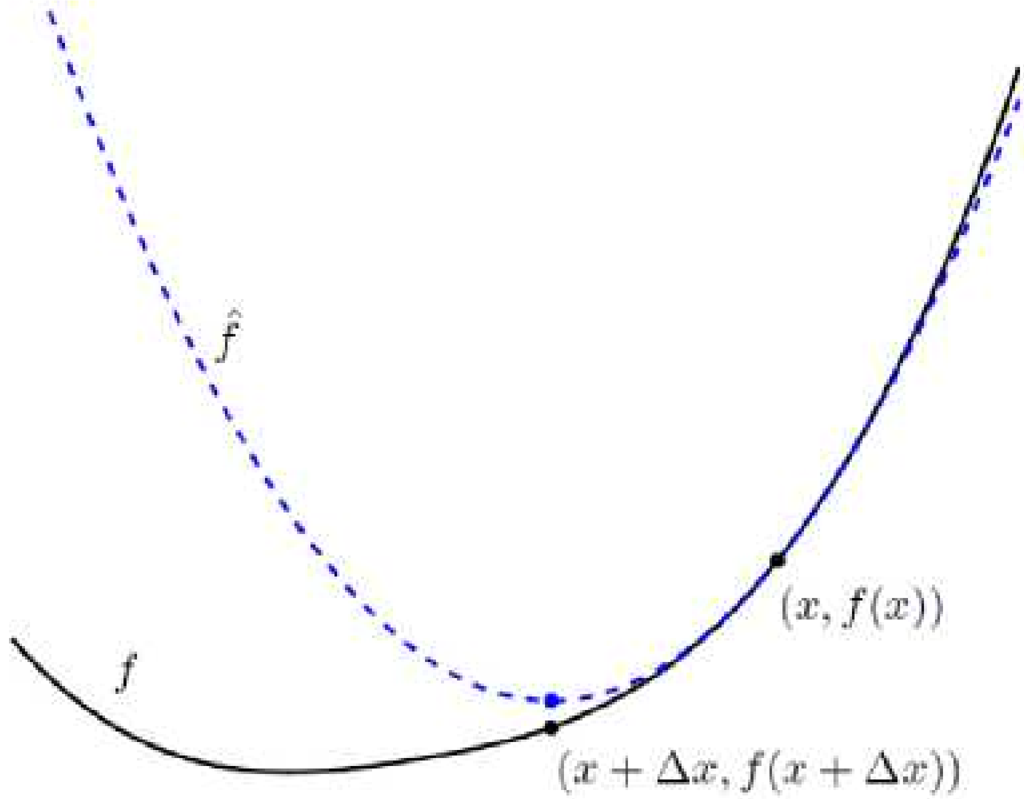
$$\Delta x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

- This is descent direction:

$$\nabla f(x)^T \Delta x = -\nabla f(x)^T [\nabla^2 f(x)]^{-1} \nabla f(x) < 0$$

Gauss-Newton's Method

$\hat{f}$ is 2-order approximation, f is true function.



Batch Gradient Descent

- Minimize empirical loss, assuming it's convex and unconstrained
  - Gradient descent on the empirical loss
  - At each step:

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial L(w, x_i, y_i)}{\partial w}\right)$$

  - Note: ate each step, gradient is the average of the gradient for all samples (i=1,…n)
  - Very slow when n is very large

Stochastic Gradient Descent

- Alternative: compute gradient from just one (or a few samples)
- Known as stochastic gradient descent:
  - At each step,

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}$$

(choose one sample i and compute gradient for that sample only)

Convex Optimization

- Unconstrained optimization
  - Gradient descent
  - Gauss-Newton’s method
  - Batch learning
  - Stochastic Gradient Descent
- Constrained optimization
  - Lagrange methods
  - Bayesian methods

Lagrange Methods

- Start with an optimization problem:

$$\arg\min_{x} f_0(x)$$

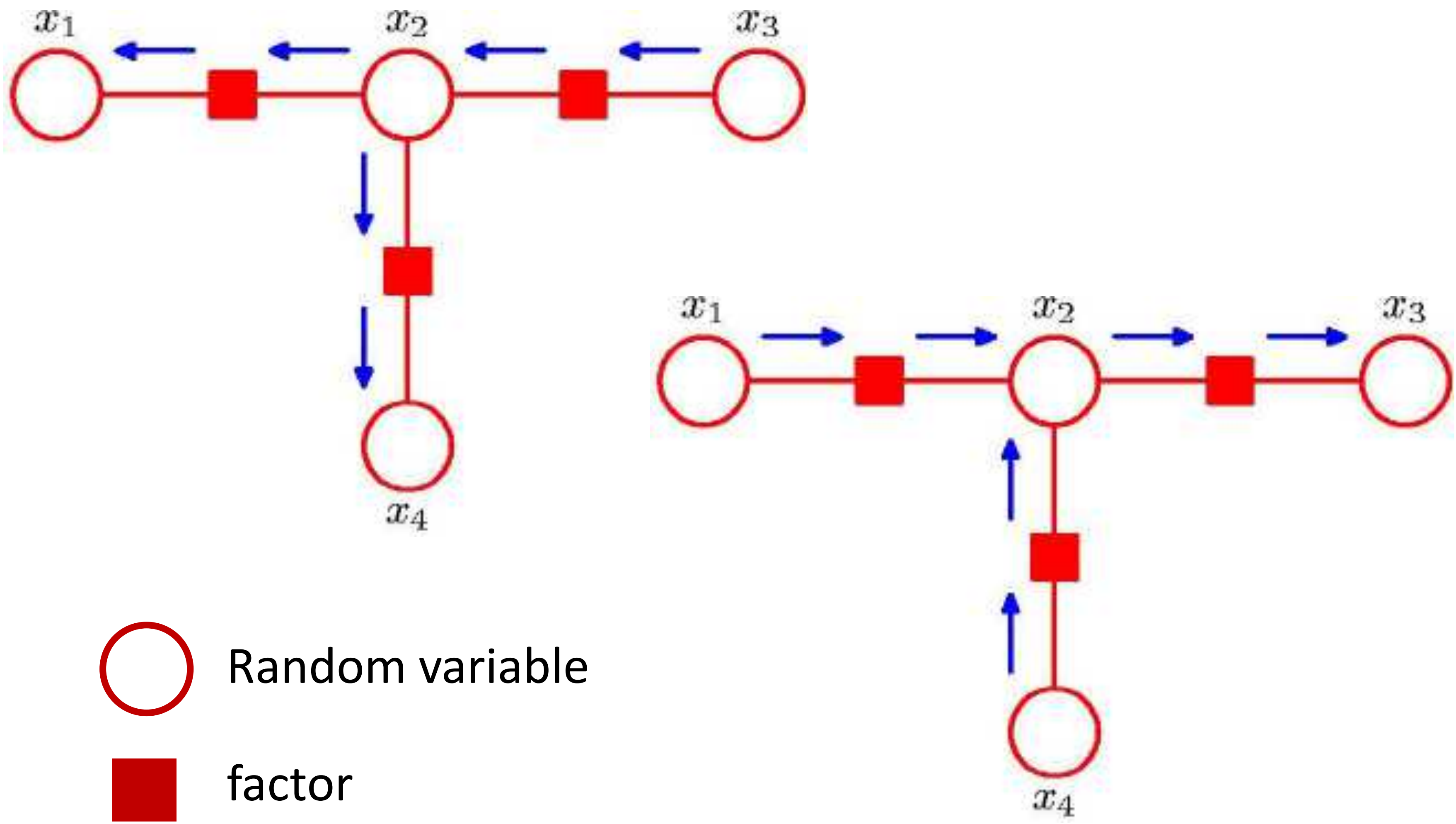
$$s.t.\, f_i(x) \leq 0, i = \{1, \dots, k\}$$

$$s.t.\, h_j(x) = 0, j = \{1, \dots, l\}$$

- Is equivalent to min-max optimization:

$$\arg\min_{x} \left[\max_{\lambda \geq 0, \gamma > 0} \left(f_0(x) + \sum_{i=1}^{k} \lambda_i f_i(x) + \sum_{j=1}^{l} \gamma_j h_j(x) \right) \right].$$

Bayesian Methods



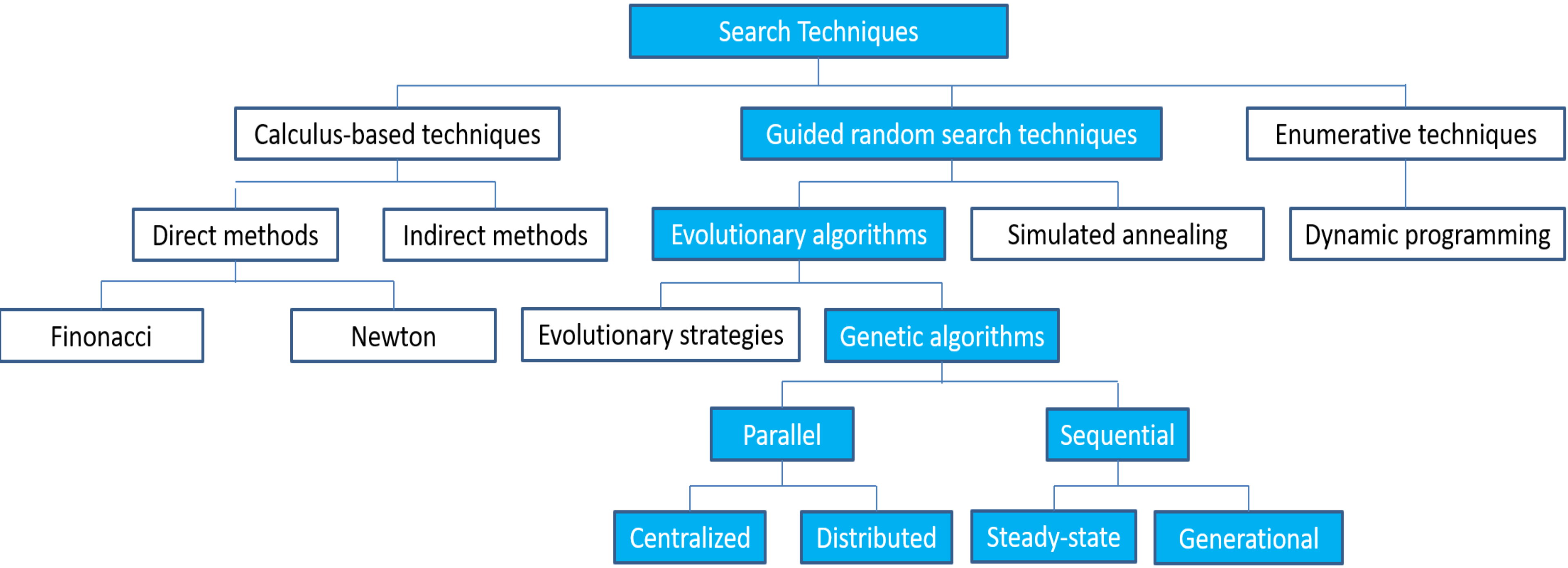
Convex Optimization for Machine Learning

- Gradient Based Methods
  - Neural networks
- Lagrange Methods
  - Support vector machines
- Bayesian Methods:
  - Expectation-Maximization methods (mixture models)
  - Variational methods (approximate models)
  - Graph optimization (belief propagation models)

Non-convex Optimization

- Convex Optimization
  - Unconstrained optimization
  - Constrained optimization
- Non-convex Optimization
  - Heuristic algorithms
  - Random search

Heuristic and Random Search



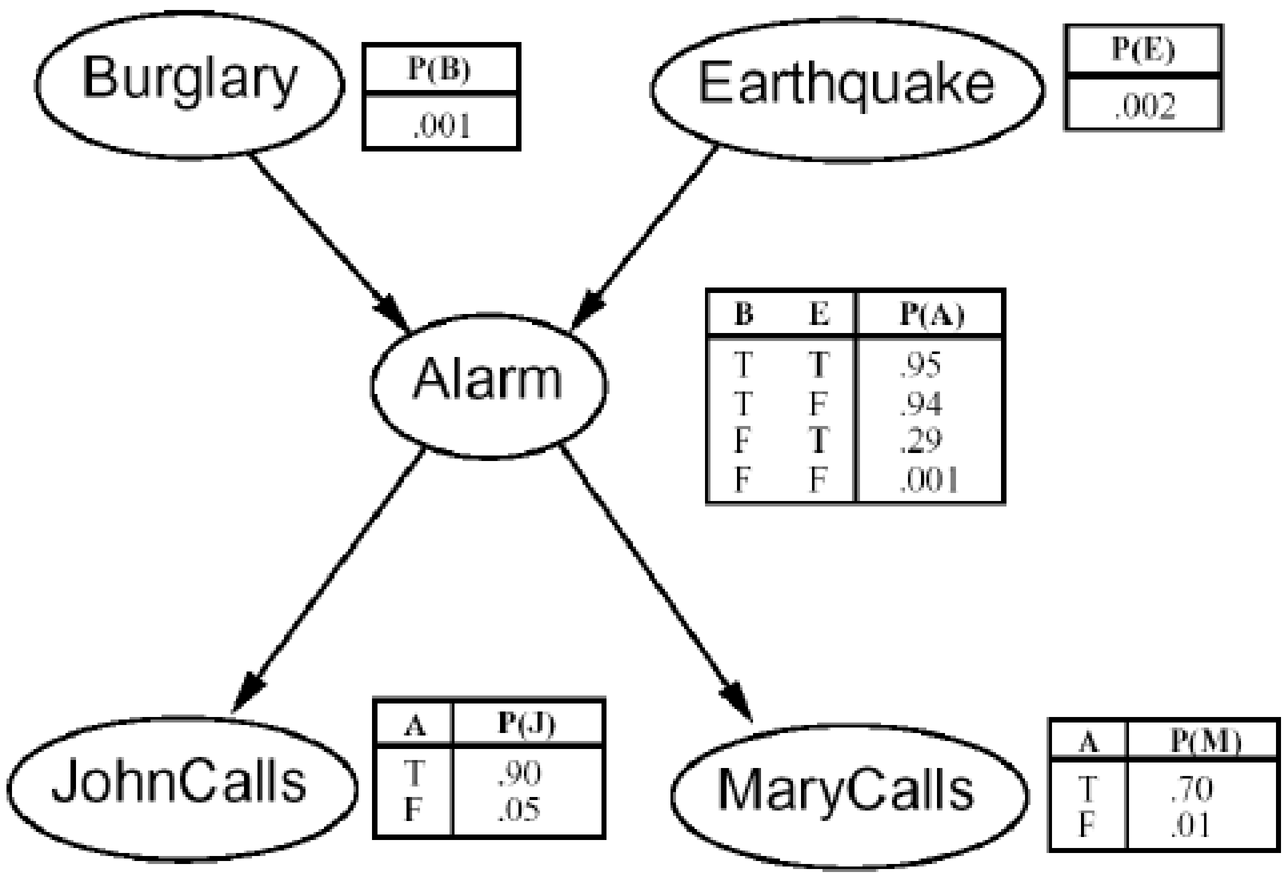
Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Algorithms

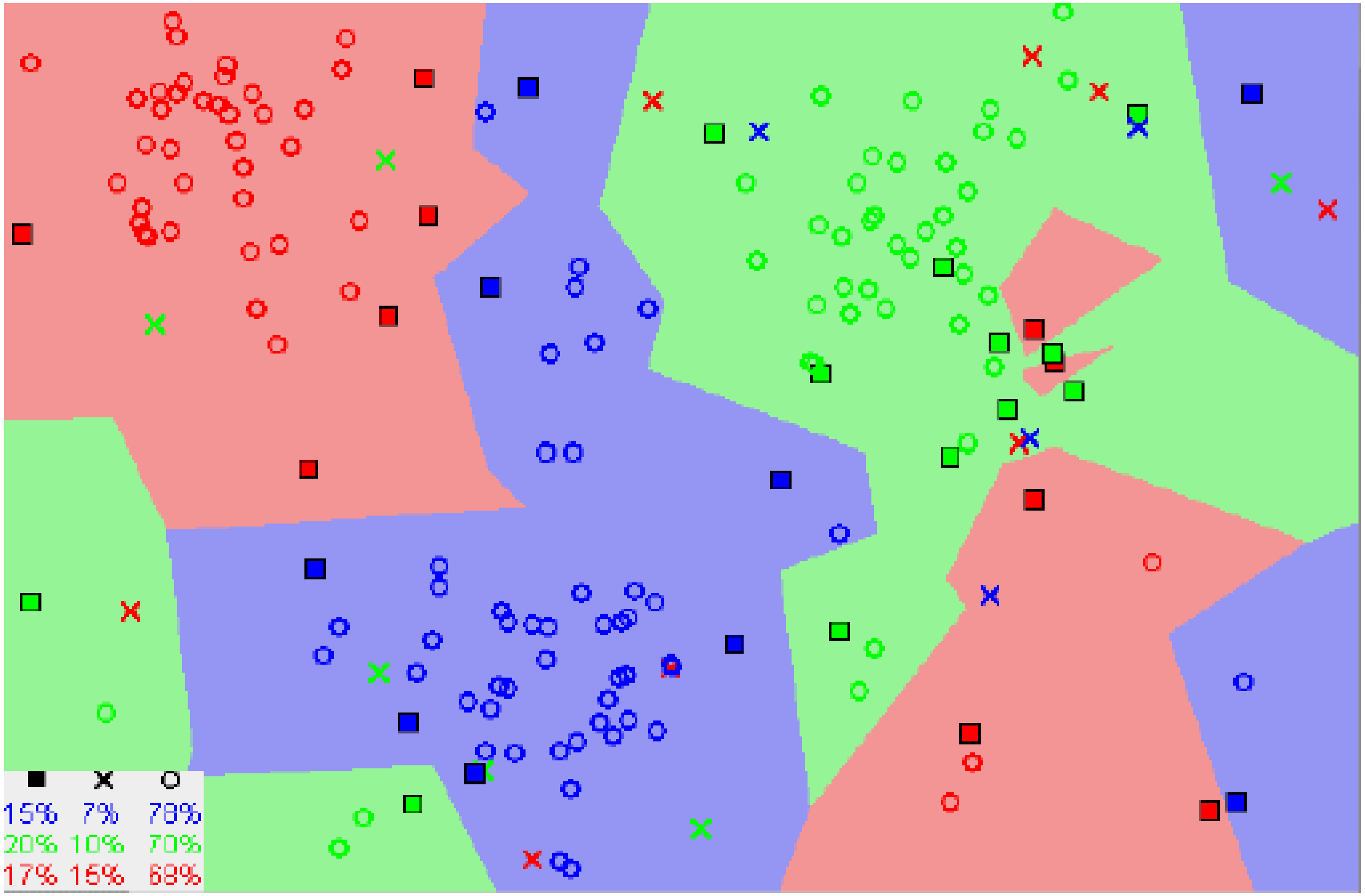
- **Bayes**
- **KNN and K-means**
- **Decision tree**
- **Support Vector Machine**
- **Boosting and Ensemble Learning**
- **Linear Statistical Learning (PCA, ICA, NMF)**
- **Nonlinear Statistical Learning (Manifold learning)**
- **Deep Neural Networks**
- **Generative Adversarial Networks**
- **Bayesian Networks**
- **Reinforcement Learning**
- **Federated Learning**

Bayes


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
T T .95
T F .94
F T .29
F F .001
JohnCalls
A P(J)
T .90
F .05
MaryCalls
A P(M)
T .70
F .01

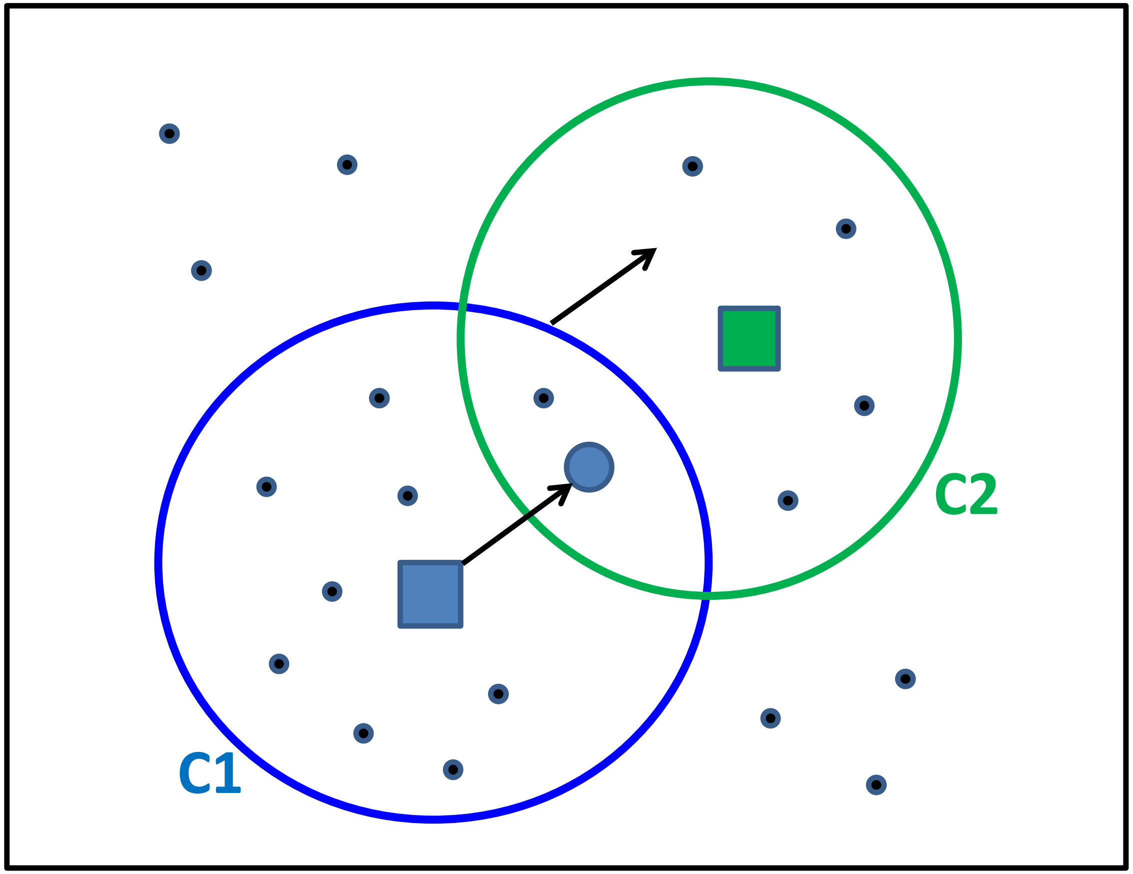
K-Nearest Neighbors

- Use training data for classification



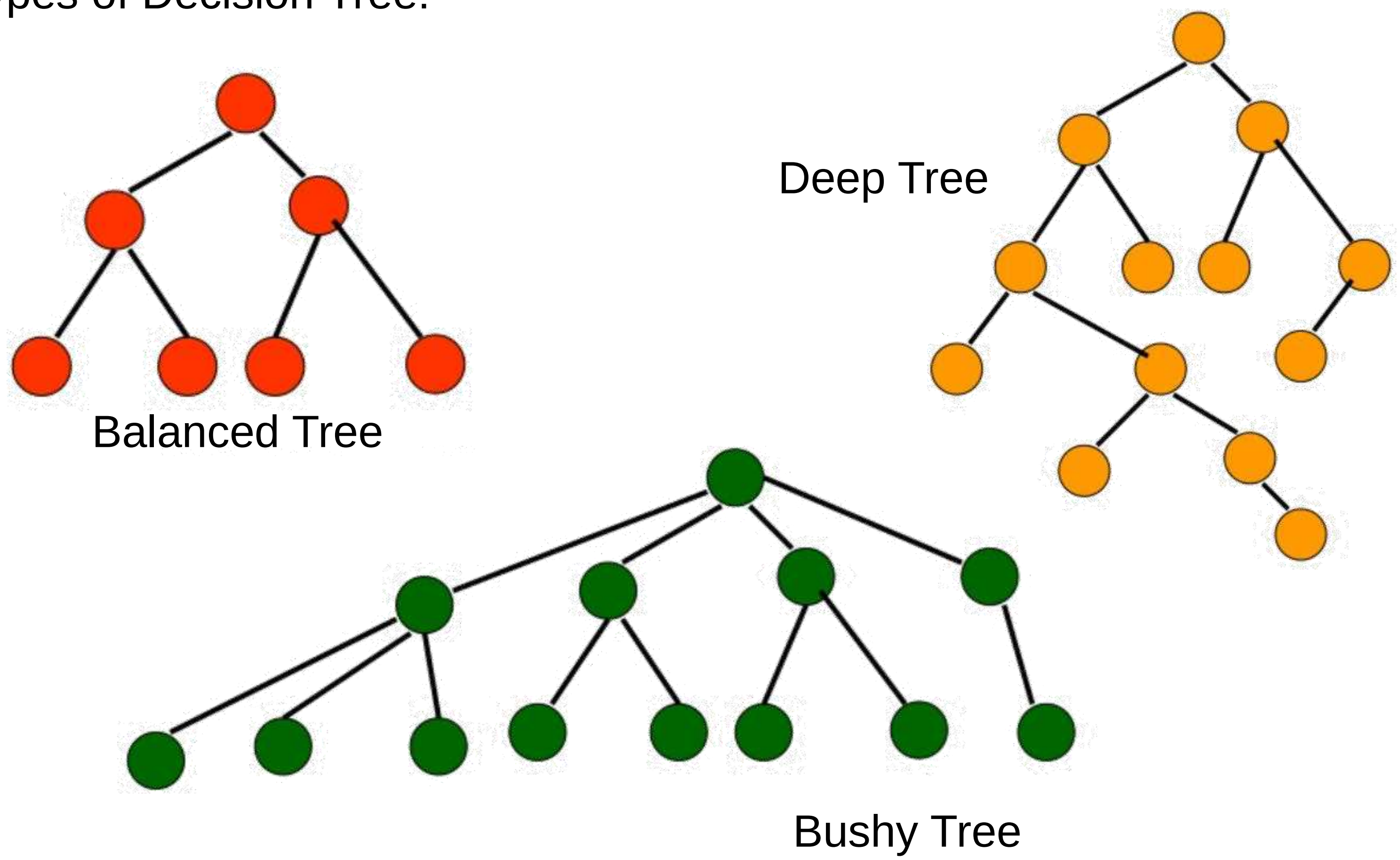
K-Means

- Mean-shift for clustering



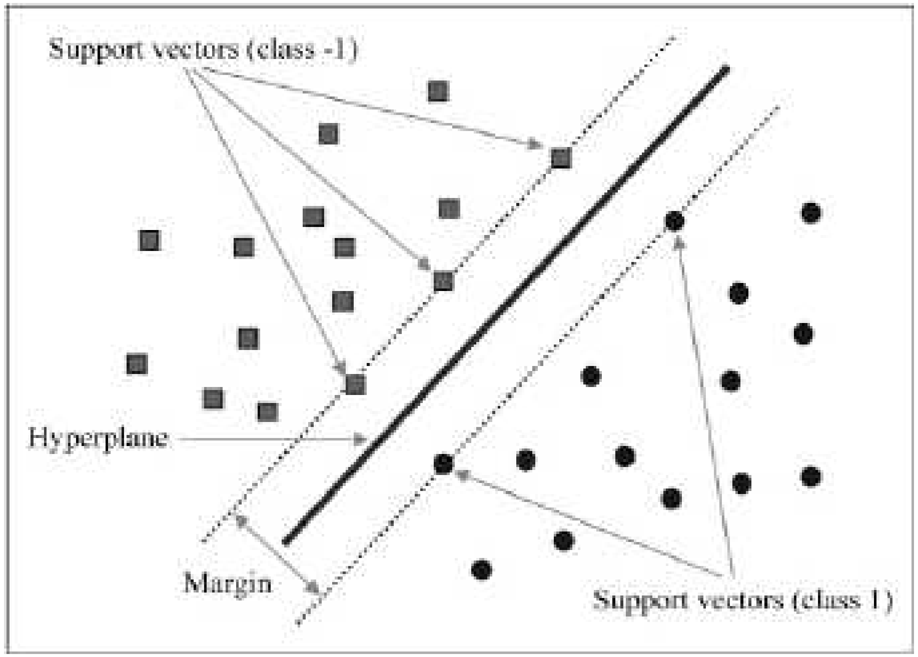
Decision Tree

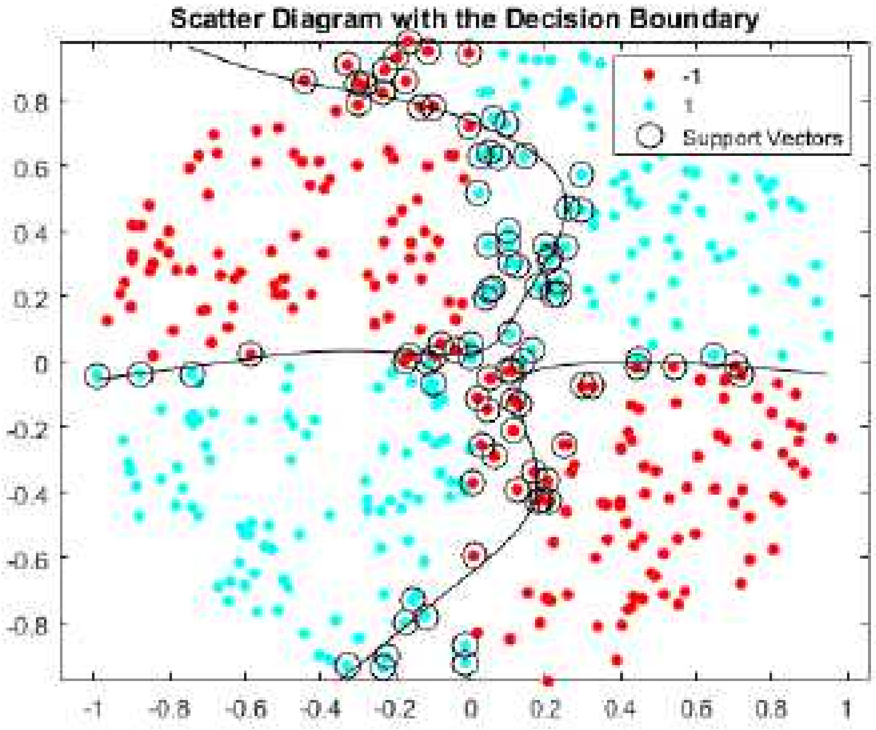
- Types of Decision Tree:



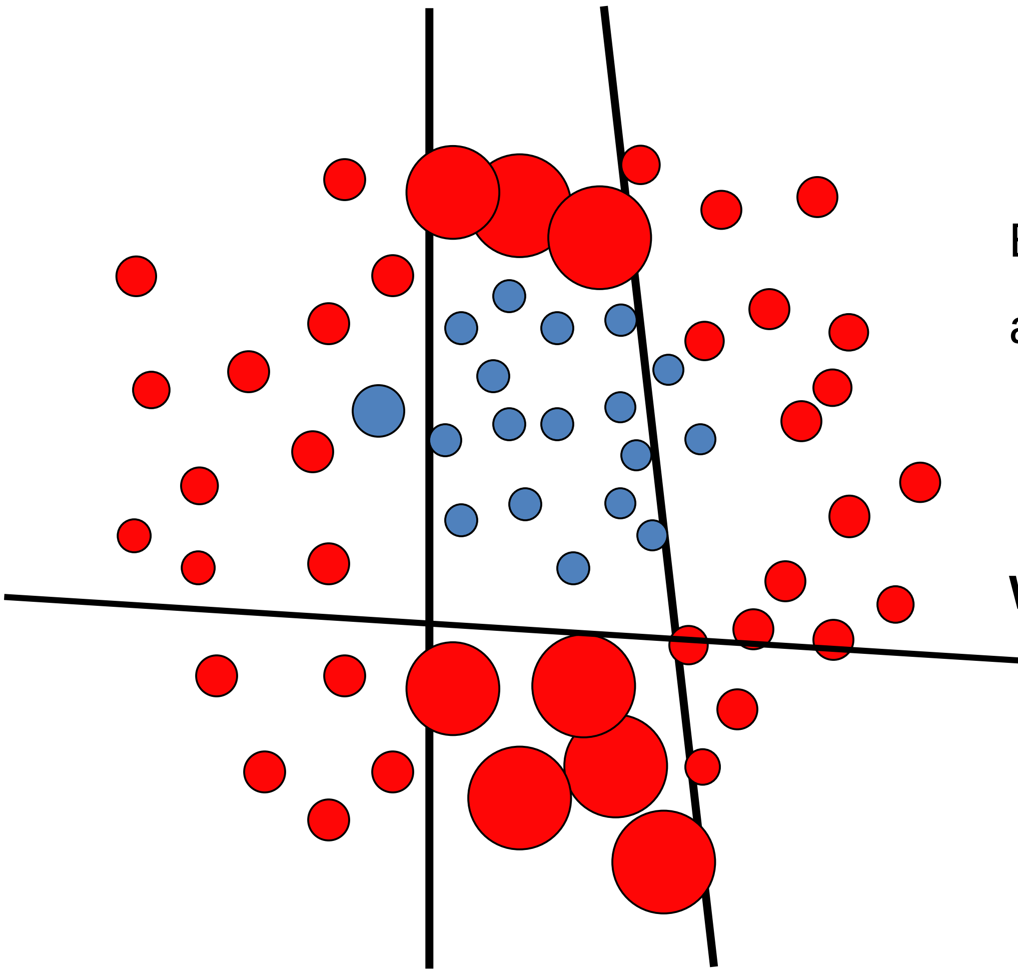
SVM

- eg. Linear SVM: $$\arg\min_{w} \sum_{i=1}^{n} ||w||^2 + C \sum_{i=1}^{n} \xi_i$$ $$\text{s.t. } 1 - y_i x_i^T w \leq \xi_i$$ $$\xi_i \geq 0$$





Boosting



Each data point has

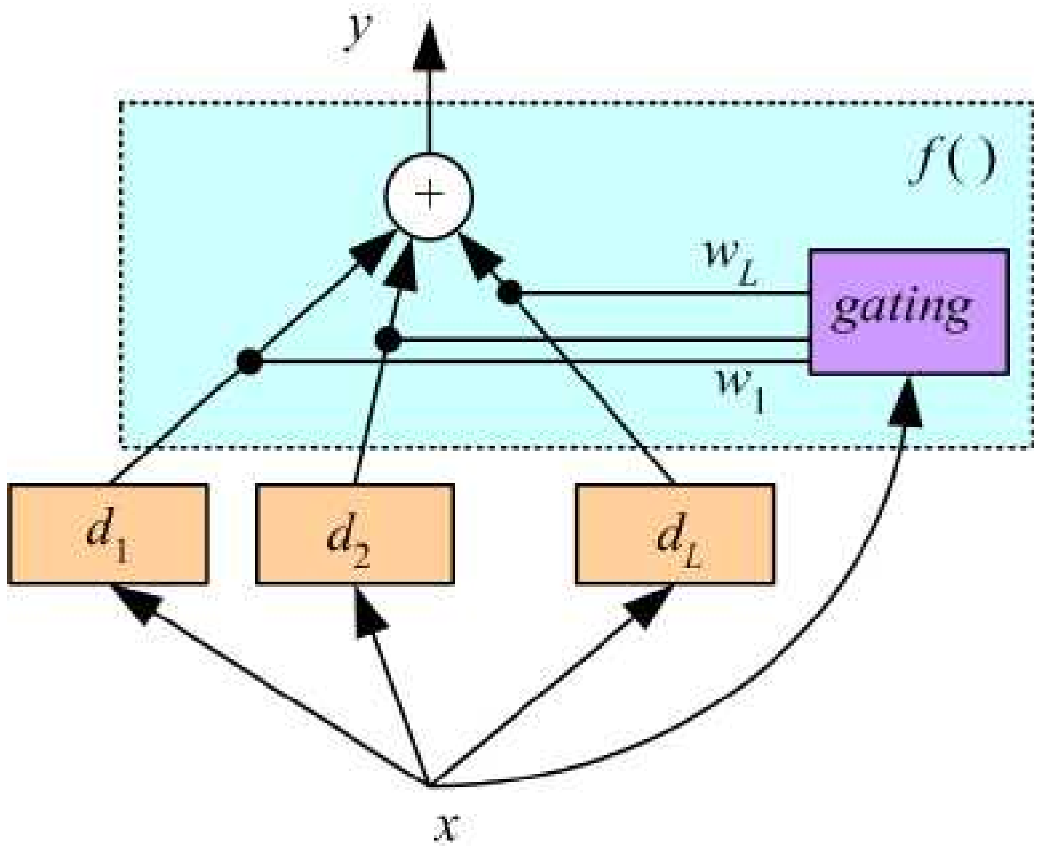
a class label:

$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{blue}) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

Ensemble Learning



$$y = \sum_{j=1}^{L} w_j d_j$$

Linear Statistical Learning

- PCA

Data, Basis, Coefficients

$$\underbrace{Y}_{\text{Data}} = \overbrace{A}^{\text{Basis}}\overbrace{X}^{\text{Coefficients}}$$

$$A_i \perp A_j$$

- ICA

Data, Mixture Coefficients, Components

$$\underbrace{Y}_{\text{Data}} = \overbrace{A}^{\text{Mixture Coefficients}}\overbrace{X}^{\text{Components}}$$

$$\min \ I(X)$$

- NMF

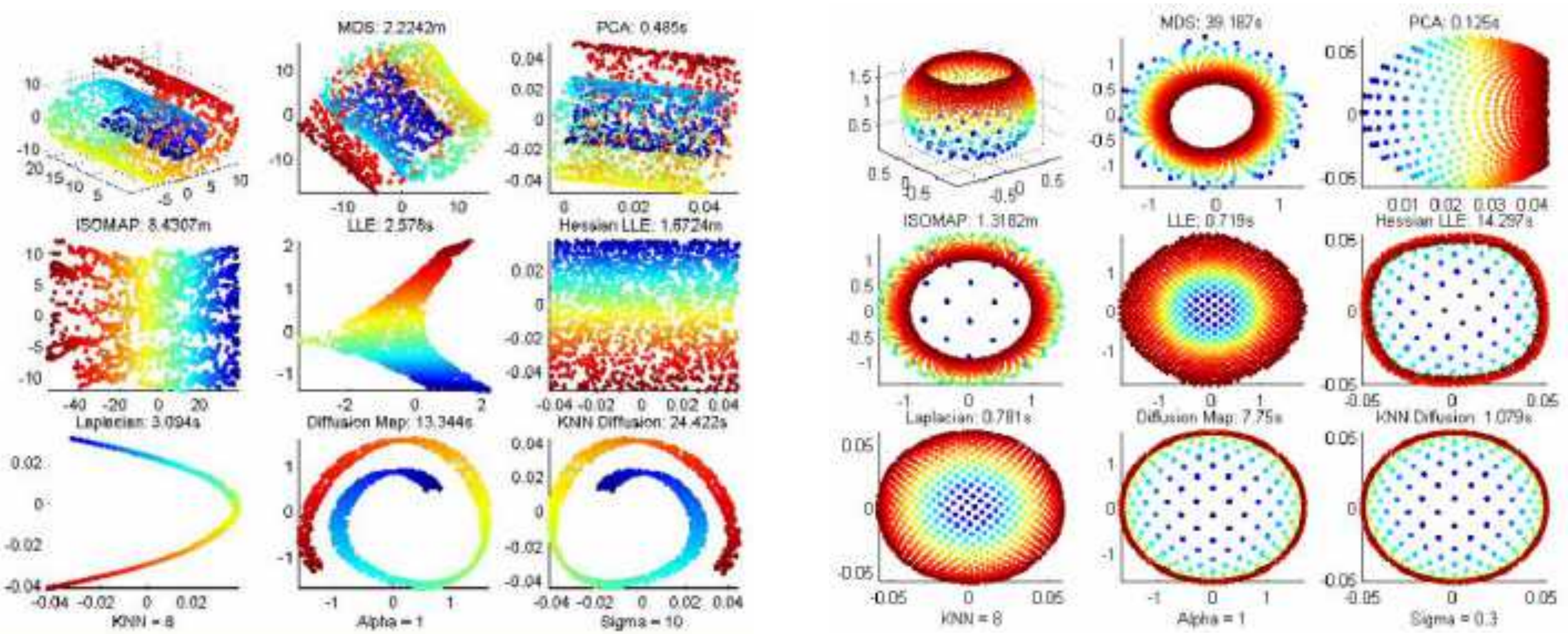
Data, Basis, Coefficients

$$\underbrace{Y}_{\text{Data}} = \overbrace{A}^{\text{Basis}}\overbrace{X}^{\text{Coefficients}}$$

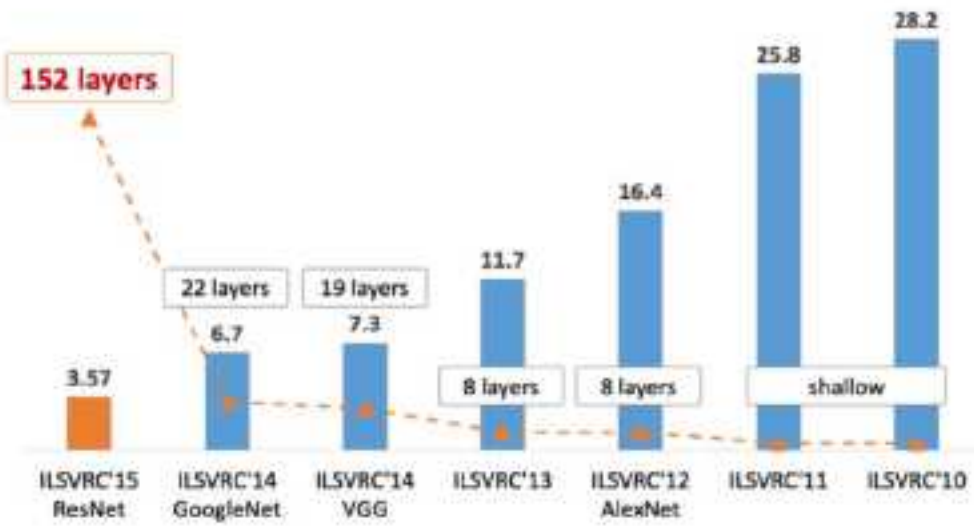
$$A, X > 0$$

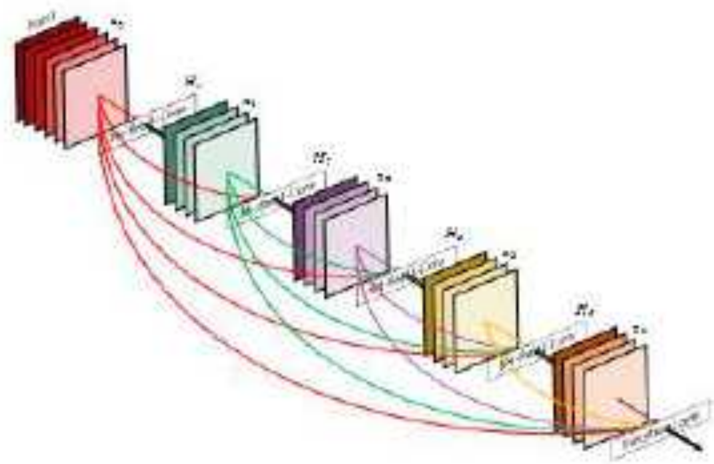
Nonlinear Statistical Learning

- Manifold learning



Deep Neural Networks

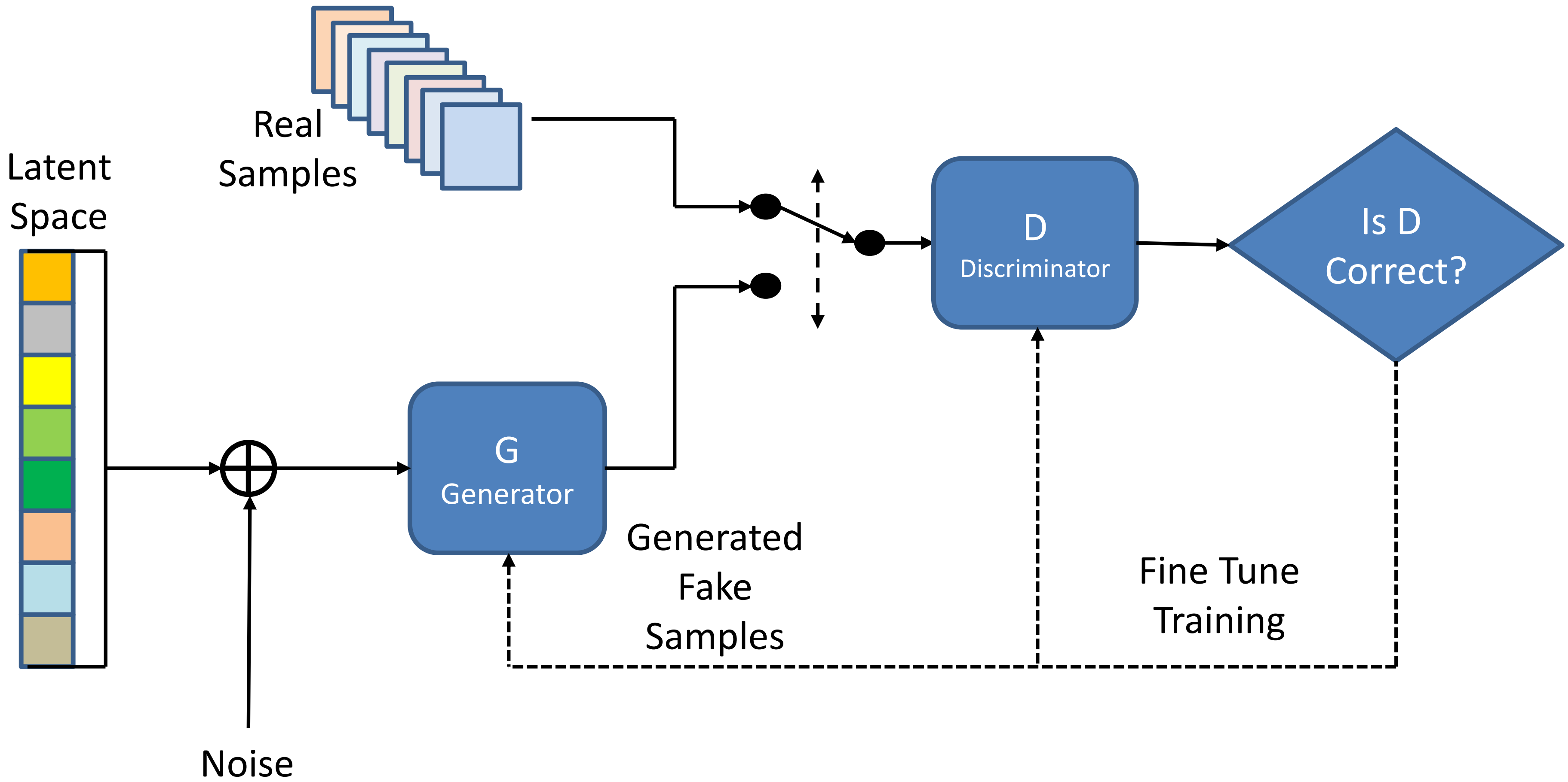




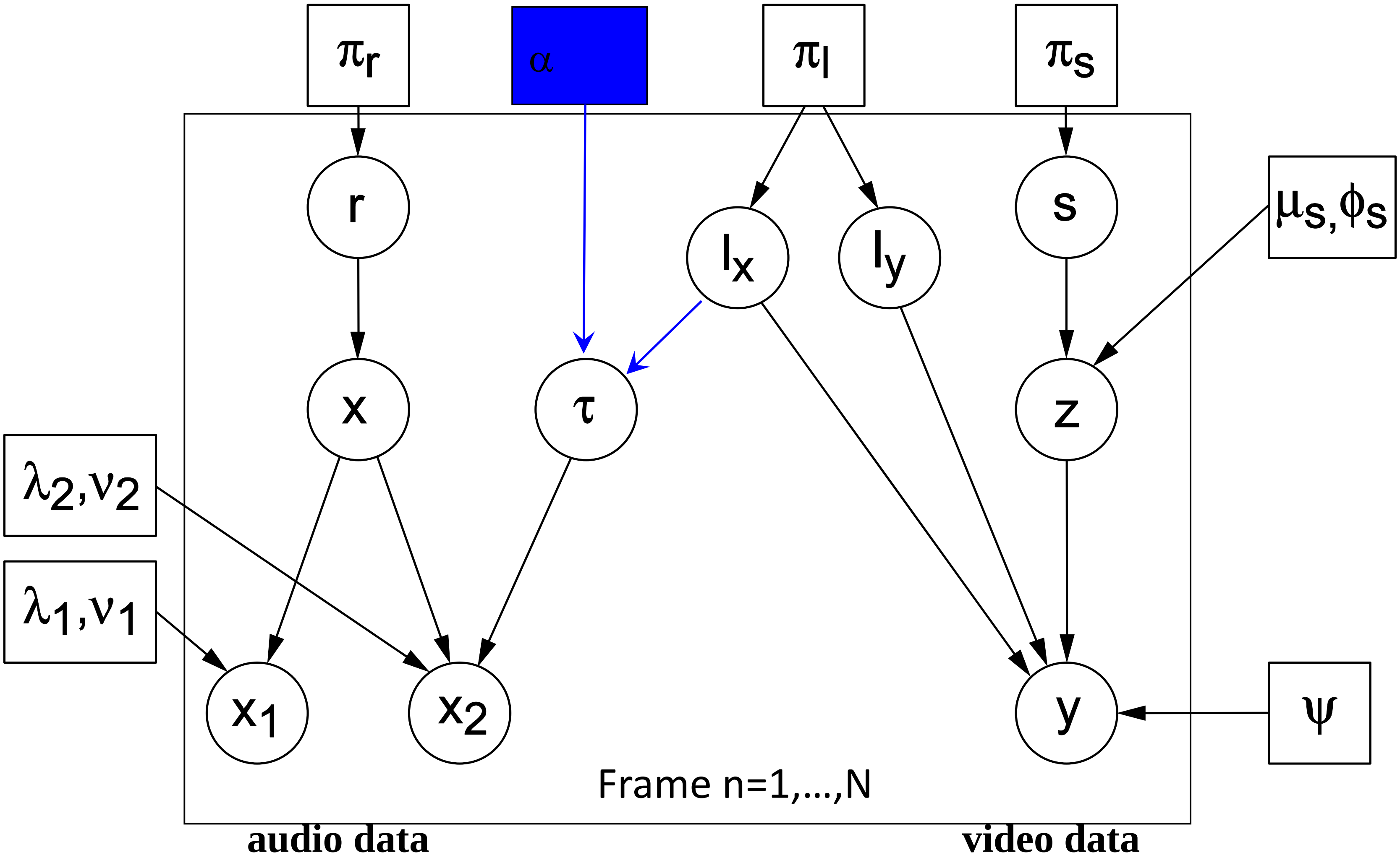
- Task： recognition
- Dataset： ILSVRC

- Huang G, Liu Z, Weinberger K Q, et al. Densely connected convolutional networks[J]. arXiv preprint arXiv:1608.06993, 2016.

Generative Adversarial Networks

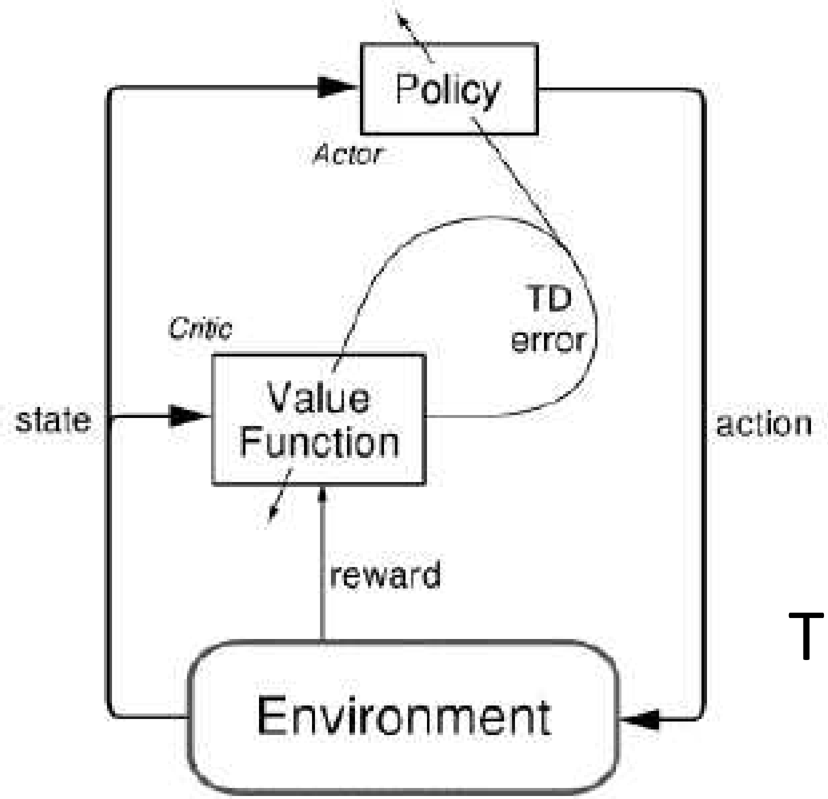


Bayesian Networks



Reinforcement Learning

- State, action, and Reward

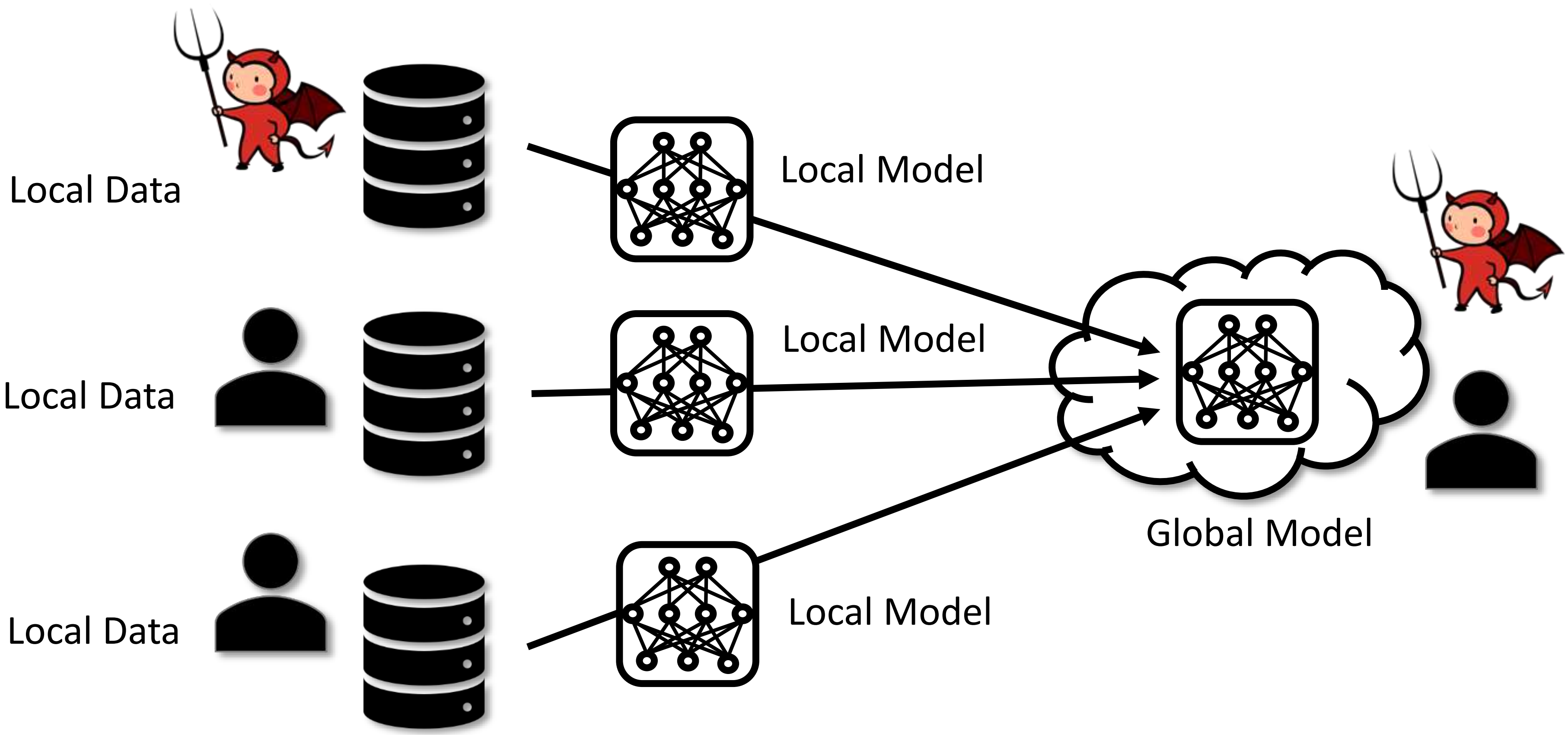


Update: Policy Function
Value Function

TD Error: Temporal Difference
between Real Reward
and Estimated Reward

Federated Learning

- Collaborative Learning



More Course Links

Stanford Machine Learning：
https://see.stanford.edu/Course/CS229/47

MIT Machine Learning： https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/index.htm

Stanford CNN for Vision： http://cs231n.stanford.edu

Stanford Deep Learning: http://cs230.stanford.edu/syllabus.html

MIT Deep Learning: http://introtodeeplearning.com/