DDA4210/AIR6002 Advanced Machine Learning
Lecture 01 Introduction and Review

Tongxin Li

School of Data Science, CUHK-Shenzhen

Spring, 2024

Overview

1. About this course
2. Review for basic machine learning methods

Logistics

- Instructor: Tongxin Li
    - Email: litongxin@cuhk.edu.cn
    - Personal website: `https://tongxin.me/`
    - Office: Daoyuan Building 323A
    - Office hours: Thu 9:50-10:50 am

Logistics

- Instructor: Tongxin Li
  - Email: litongxin@cuhk.edu.cn
  - Personal website: `https://tongxin.me/`
  - Office: Daoyuan Building 323A
  - Office hours: Thu 9:50-10:50 am

SCAN ME

- Course website: `https://tongxin.me/DDA4210/`

Logistics

Teaching assistants:

- Fangchen Yu, 220019040@link.cuhk.edu.cn
- Yifei Wu, 223040255@link.cuhk.edu.cn
- Fanzeng Xia, 223040232@link.cuhk.edu.cn

USTFs:

- Huihan Yang, 120090438@link.cuhk.edu.cn
- Yiming Xiong, 120090721@link.cuhk.edu.cn
- Rongxiao Qu, 120020144@link.cuhk.edu.cn
- Jiayi Yao, 120040070@link.cuhk.edu.cn

Assessment (DDA4210)

- Homework (30%)
  - Three assignments (tri-weekly)
  - Involves theory, analysis, computation, and programming.

Assessment (DDA4210)

- Homework (30%)
  - Three assignments (tri-weekly)
  - Involves theory, analysis, computation, and programming.
- Course project (35%)
  - Format: Python programming for advanced machine learning
  - Topic: determined by yourself (given a few examples or choices)
  - Teamwork: 1 to 4 members per team
  - Outcome evaluation: [report(25%)+presentation(75%)] $\times r_{\text{ind}}$
    - presentation: $75\% = 10\%\text{peer} + 25\%\text{TA} + 40\%\text{instructor}$
    - $r_{\text{ind}} \in \{0.5, 0.8, 1\}$: it is rated by your teammates on your contribution; 0.5 (or 0.8) means your contribution is less than 20% (or 50%) of the expected workload.
  - Evaluation criteria: significance, novelty, technical soundness, completeness

Assessment (DDA4210)

- Homework (30%)
  - Three assignments (tri-weekly)
  - Involves theory, analysis, computation, and programming.
- Course project (35%)
  - Format: Python programming for advanced machine learning
  - Topic: determined by yourself (given a few examples or choices)
  - Teamwork: 1 to 4 members per team
  - Outcome evaluation: [report(25%)+presentation(75%)]$\times r_{\text{ind}}$
    - presentation: $75\% = 10\%\text{peer} + 25\%\text{TA} + 40\%\text{instructor}$
    - $r_{\text{ind}} \in \{0.5, 0.8, 1\}$: it is rated by your teammates on your contribution; 0.5 (or 0.8) means your contribution is less than 20% (or 50%) of the expected workload.
  - Evaluation criteria: significance, novelty, technical soundness, completeness
- Final exam (35%)
  - Single-choice questions, calculation, math derivation/proofs, etc.

Assessment (AIR6002)

- Homework (40%)
  - Three assignments (tri-weekly)
  - Involves theory, analysis, computation, and **more** programming.

Assessment (AIR6002)

- Homework (40%)
  - Three assignments (tri-weekly)
  - Involves theory, analysis, computation, and **more** programming.
- Course project (60%)
  - Format: Cutting-edge topics in advanced machine learning
  - Topic: determined by yourself (given a few examples or choices)
  - Teamwork: 1 to 3 members per team
  - Outcome evaluation: [mid-term proposal(10%)+ report(25%)+presentation(65%)] $\times r_{\text{ind}}$
    - presentation: $75\% = 10\%\text{peer} + 25\%\text{TA} + 40\%\text{instructor}$
    - $r_{\text{ind}} \in \{0.5, 0.8, 1\}$: it is rated by your teammates on your contribution; 0.5 (or 0.8) means your contribution is less than 20% (or 50%) of the expected workload.
  - Evaluation criteria: significance, novelty, technical soundness, completeness

Some remarks

- Plagiarism violates the university policy of “**Academic Integrity**"
  - Plagiarism in homework assignments, course projects, and final exam will be dealt with **severity**.
  - For example, assignments with plagiarism will be graded as zero.
  - Repeated plagiarism will lead to an "F" for the entire course.
- Attendance requirement
  - Attending lectures/tutorials onsite is highly encouraged.
  - Please answer or raise questions actively.
  - Let the instructor/TAs be able to recognize you as a student in the class.
- Participation in Course&Teaching Evaluation (CTE)
  - Your feedback (either positive or negative) helps improve the course and make it even better.

Some remarks

*The building blocks of machine learning are **data**, **models**, and **algorithms**.*

The building blocks of advanced machine learning (DDA4210/AIR6002) are more complicated data, more powerful models, and state-of-the-art algorithms in real-world applications.

modern

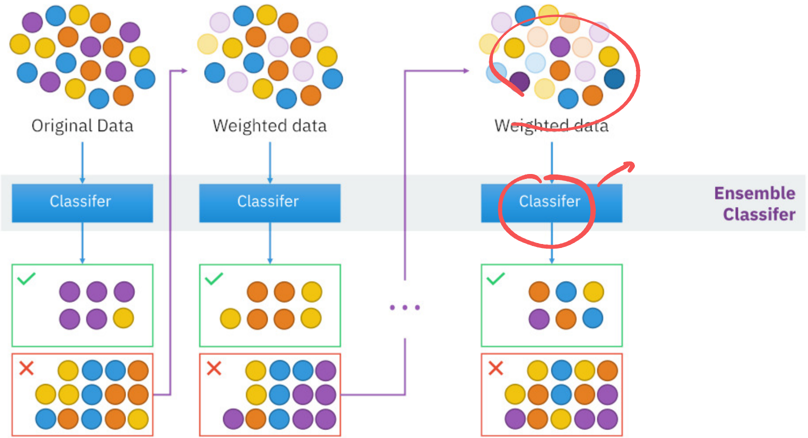
Syllabus

1. Review of basic machine learning methods
2. Advanced ensemble learning
3. Learning theory
4. Advanced applications: recommendation and search
5. Spectral clustering and semi-supervised learning
6. Graph neural networks
7. Nonlinear dimensionality reduction and data visualization
8. Generative models (VAE, GAN, diffusion model)
9. Causal machine learning
10. Privacy in machine learning
11. Fairness in machine learning
12. Interpretability in machine learning
13. Course project presentation
14. Final exam

Basic machine learning methods

- Linear regression and classification
- K-nearest neighbor method
- Decision tree, bagging, and random forest
- Support vector machine
- Neural networks (MLP, CNN, and RNN)
- K-means and Gaussian mixture models
- Principal component analysis

Advanced Machine Learning: Boosting

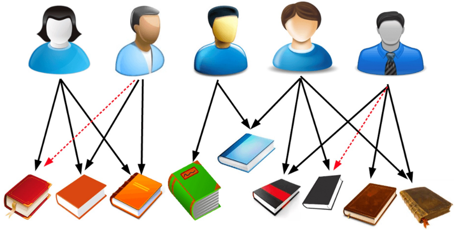


Boosting is an ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones.—-Wikipedia

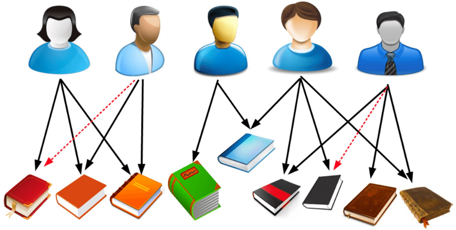
Machine Learning Theory[1]

- Also known as *Computational Learning Theory*
- Aims to understand the fundamental principles of learning as a computational process and combines tools from Computer Science and Statistics
  - Creating mathematical models that capture key aspects of machine learning, in which one can analyze the inherent ease or difficulty of different types of learning problems.
  - Proving guarantees for algorithms (under what conditions will they succeed, how much data and computation time is needed) and developing machine learning algorithms that provably meet desired criteria.
  - Mathematically analyzing general issues, such as: "When can one be confident about predictions made from limited data?", "What kinds of methods can learn even in the presence of large quantities of distracting information?"

[1] https://www.cs.cmu.edu/~avrim/Talks/mlt.pdf

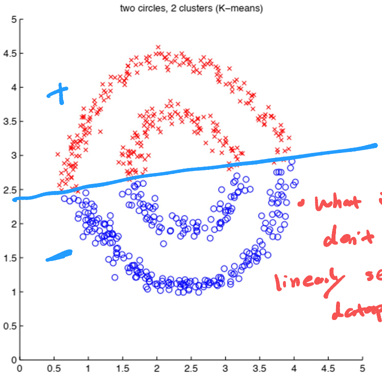



BIG
SALE

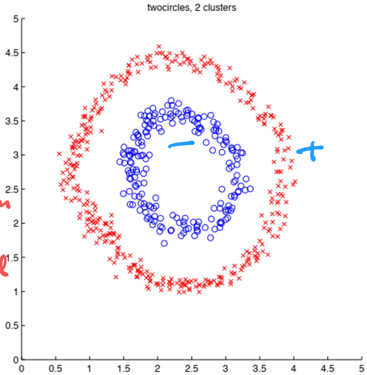


- Collaborative filtering methods
- Content-based methods
- Hybrid methods

K-means



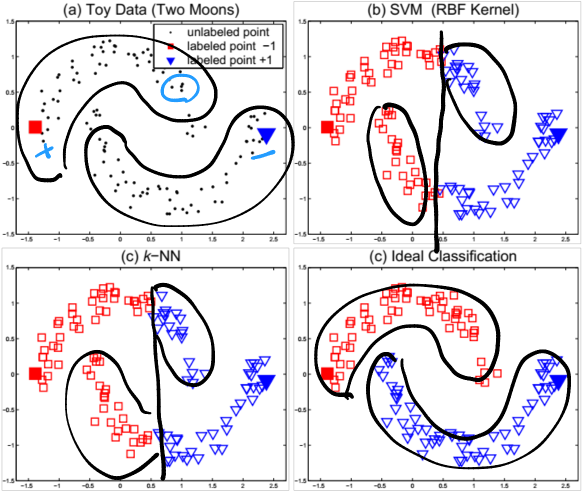
Spectral clustering



• what if you don't have linearly separated datapts?

Why Semi-Supervised Learning?

Classification on the two moons pattern [Zhou et al. 04]:
(a) two labeled points; (b) SVM with an RBF kernel; (c) k-NN with k = 1.



Graph-Structured data cannot be well handled by conventional neural networks!

GNN

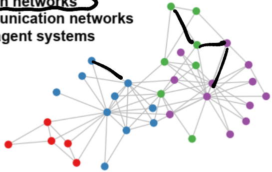
$X = \{x_i\}_{i=1}^{n}$

$G = (V, E)$

$\{x_1, \ldots, x_n\}$

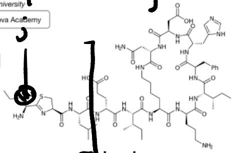
A lot of real-world data does not "live" on grids

Social networks
Citation networks
Communication networks
Multi-agent systems

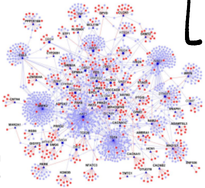


Knowledge graphs

i



Molecules



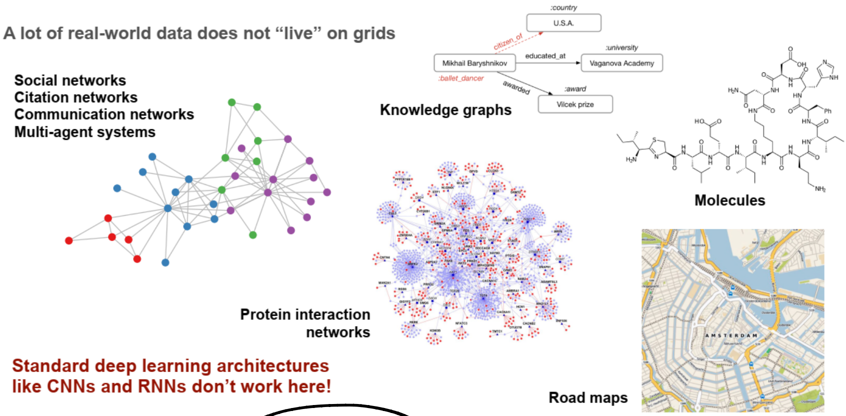
Protein interaction networks

Road maps

Standard deep learning architectures like CNNs and RNNs don't work here!

- classification
- prediction.

Graph-Structured data cannot be well handled by conventional neural networks!



Tasks: node classification, link prediction, graph classification

The image is from Thomas Kipf.

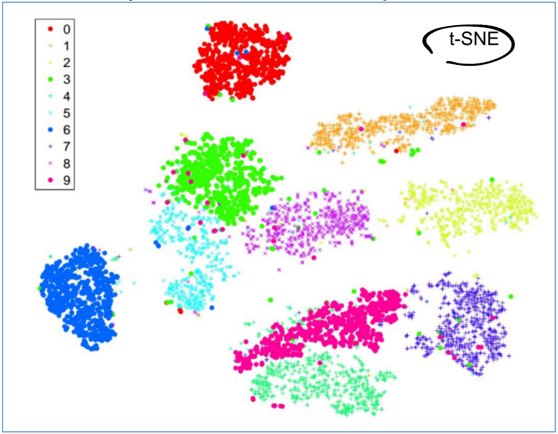
NLDR: nonlinear Dimensionality reduction

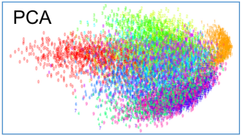
Example: visualizing MNIST handwritten digits (10 classes)

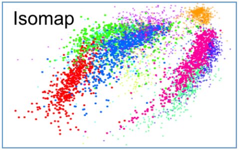
$\boxed{1}\ \boxed{2}\ \boxed{1} \cdots \boxed{9} \cdots$

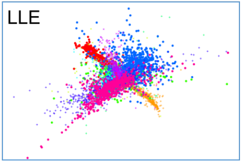
7 1

$\{0, 1, 2, 3 \cdots 9\}$







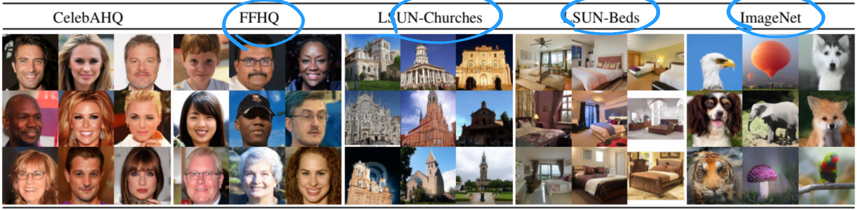


Advanced Machine Learning: Generative Models

VAE, GAN. Diffusion model.

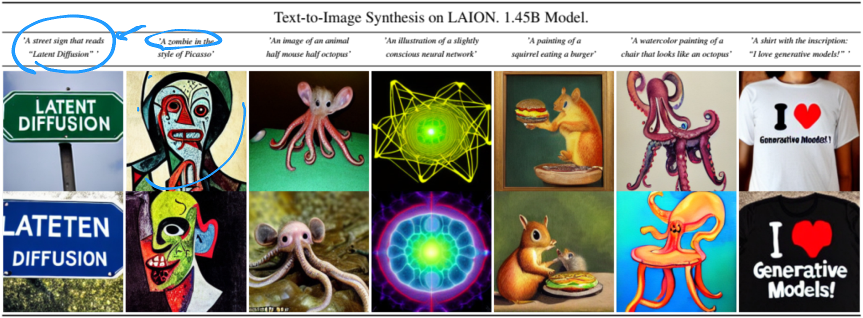
Use the model trained on training data to generate new data, such as images, text, audio, and videos.

Generating images [Rombach et al. 2022]



Use the model trained on training data to generate new data, such as images, text, audio, and videos.

Text to image [Rombach et al. 2022]



First PART.

1. Ensemble learning.
2. Recommendation system.
3. Clustering.
4. Nonlinear DR.
5. GNN
6. Generative Models.
7. Learning theory

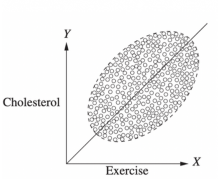
- Develop ML models and algorithms that are accurate.

Second PART.

Next, besides accuracy.

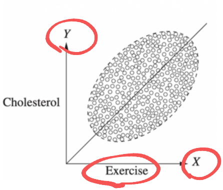
- explainable.
- fair
- privacy-preserving.
- safe/robust.

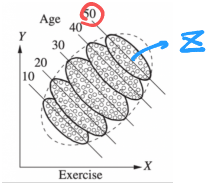
- Causal inference

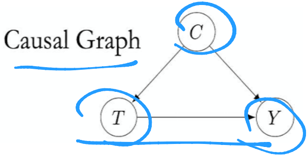


Advanced Machine Learning: Causal Learning

- Causal inference

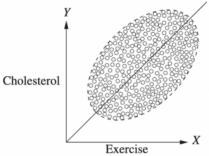


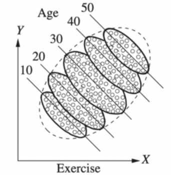


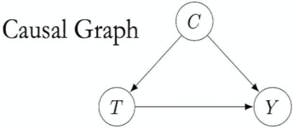


Advanced Machine Learning: Causal Learning

- Causal inference



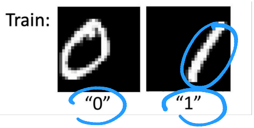


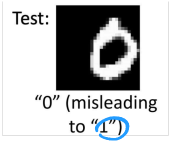


- Causal machine learning (for non-i.i.d problem)

$X = (x_1, \dots x_n)$

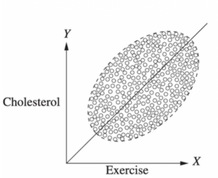
$x_i \sim P$ iid.

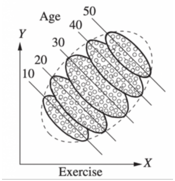


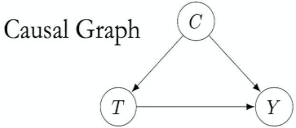


Advanced Machine Learning: Causal Learning

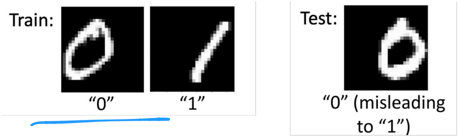
- Causal inference



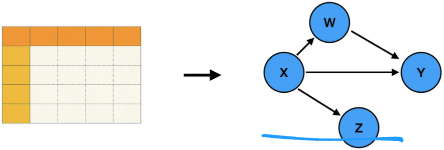




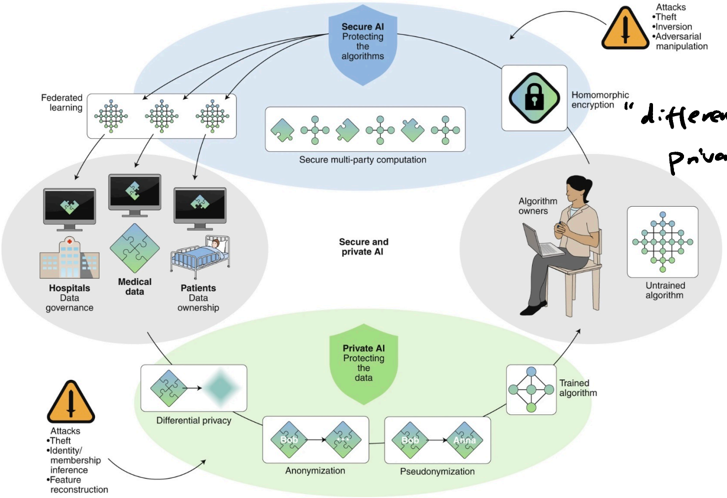
- Causal machine learning (for non-i.i.d problem)



- Causal discovery



Schematic overview of the relationships and interactions between data, algorithms, actors and techniques in the field of secure and private AI [Kaissis et al. 2020]:

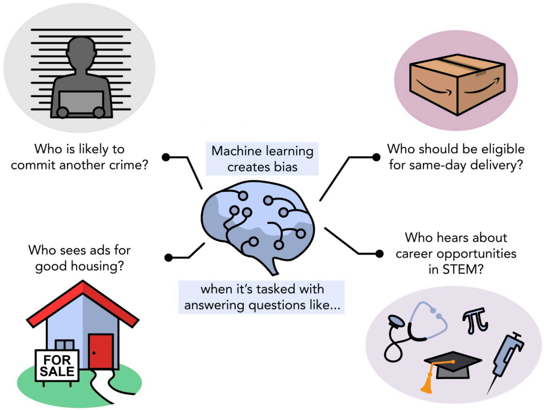


"differential privacy"

Advanced Machine Learning: Fairness

Where does the unfairness in machine learning algorithms come from? How can we address the unfairness?

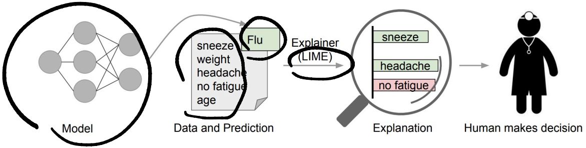
Examples of how bias in machine learning can affect our daily lives [Grabski et al. 2020]:



Advanced Machine Learning: Interpretability

Understanding the reasons behind decisions made by black-box machine learning models

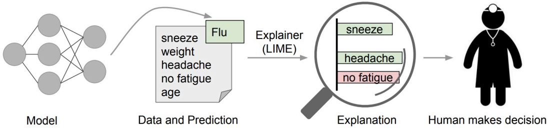
Explaining individual outputs of a model that predicts that a patient has the flu [Ribeiro et al. 2016]:



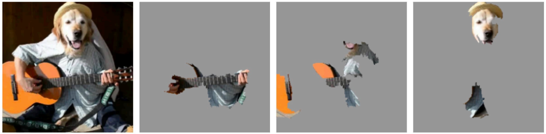
Advanced Machine Learning: Interpretability

Understanding the reasons behind decisions made by black-box machine learning models

Explaining individual outputs of a model that predicts that a patient has the flu [Ribeiro et al. 2016]:



Explaining an image classification prediction made by Google's Inception neural network [Ribeiro et al. 2016]:



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Review for basic machine learning methods

- Linear regression and classification
- K-nearest neighbor method
- Decision tree, bagging, and random forest
- Support vector machine
- Neural networks (MLP, CNN, and RNN)
- K-means and Gaussian mixture models
- Principal component analysis

supervised Methods

unsupervised.

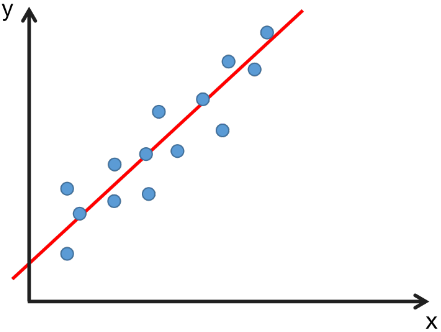
Review: Linear Regression

- Training data: $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots (\mathbf{x}_N, \mathbf{y}_N)\}$
  - $\mathbf{x}_i \in \mathbb{R}^D$, $\mathbf{y}_i \in \mathbb{R}^K$, $i = 1, 2, \dots, N$
  - with i.i.d assumption usually

$N >> D$

$N >> K$

- Learn a linear function $f_{\mathbf{W},\mathbf{b}}(\mathbf{x}) = \underbrace{\mathbf{W}\mathbf{x} + \mathbf{b}}_{\text{affine.}}$ from $\mathcal{D}$
  - $\mathbf{W} \in \mathbb{R}^{K \times D}$, $\mathbf{b} \in \mathbb{R}^K$



Review: Linear Regression

putting $\{y_i\}, \{x_i\}$ here

$\begin{cases} x_i \in \mathbb{R}^D \\ y_i \in \mathbb{R}^K \\ N := \#\text{ of samples.} \end{cases}$

- Linear regression (least squares)

(1) $\sum_{i=1}^{N} \|y_i - \bar{w}\bar{x}_i\|^2 = \|Y - \bar{W}\bar{X}\|_F^2$

$\bar{w} := [b \ \ w]$

$\bar{x}_i := \begin{bmatrix} 1 \\ x_i \end{bmatrix}$

$$\min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{W}\mathbf{x}_i - \mathbf{b}\|_2^2 \tag{1}$$

Q: • Space complexity ?

$O(N(K+D))$ $(N >> D)$.

- Ridge regression

$$\min_{\mathbf{W},\mathbf{b}} \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{W}\mathbf{x}_i - \mathbf{b}\|^2 + \underbrace{\frac{\lambda}{2}\|\mathbf{W}\|_F^2}_{\text{regularizer.}} \tag{2}$$

Avoid overfitting

- LASSO $\bar{w} = Y\bar{X}^T(\bar{X}\bar{X}^T)^{-1}$

"normal equation"

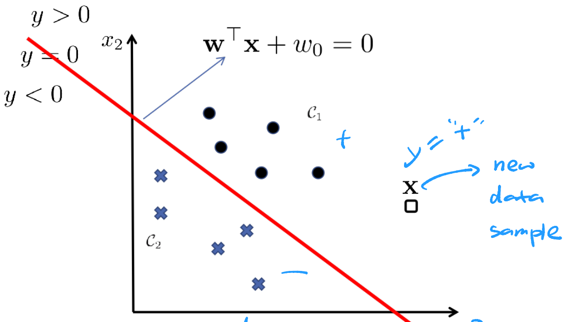
$$\min_{\mathbf{W},\mathbf{b}} \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{W}\mathbf{x}_i - \mathbf{b}\|^2 + \underbrace{\lambda\|\mathbf{W}\|_1}_{\text{enforces sparsity.}} \tag{3}$$

Q: ℓ_0-norm ?

* $\|\mathbf{W}\|_F = \sqrt{\sum_i \sum_j w_{ij}^2}$, $\|\mathbf{W}\|_1 = \sum_i \sum_j |w_{ij}|$

Review: Linear Classification

- Training data: $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_N, y_N)\}$
  - $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \{+1, -1\}$, $i = 1, 2, \ldots, N$
  - with i.i.d assumption usually
- Learn a linear classifier $f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ from $\mathcal{D}$



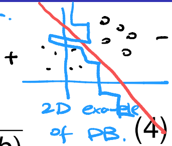
Issue: We need a binary output ?

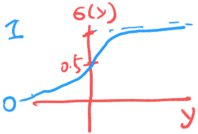
Idea: "probability"

Review: Linear Classification

Q: Decision Boundary of LR: Linear Non-linear.

- Logistic regression (binary classification, $y \in \{0, 1\}$)





σ(y)

$$f_{\mathbf{w},b}(\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x} - b)} \tag{4}$$

$\begin{cases} \text{probability } \sigma(w^T x+b) \rightarrow y\in\{1\} \\ \text{probability } 1-\sigma(w^T x+b) \rightarrow y\in\{0\} \end{cases}$ affine

$$\min_{\mathbf{w},b} \; -\frac{1}{N}\sum_{i=1}^{N} \left(y_i \log f_{\mathbf{w},b}(\mathbf{x}_i) + (1 - y_i)\log(1 - f_{\mathbf{w},b}(\mathbf{x}_i))\right) \tag{5}$$

cross-entropy

- Softmax regression (multi-class classification, $\mathbf{y} \in \{0, 1\}^K$)

$$f^{(j)}_{\mathbf{W},\mathbf{b}}(\mathbf{x}) = \frac{\exp(\mathbf{w}_j^\top \mathbf{x} + b_j)}{\sum_{c=1}^{K} \exp(\mathbf{w}_c^\top \mathbf{x} + b_c)} \tag{6}$$

$$\min_{\mathbf{w},\mathbf{b}} \; -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} \log f^{(j)}_{\mathbf{w},b}(\mathbf{x}_i) \tag{7}$$

Review: K-Nearest Neighbor Method

- k-NN: a nonlinear regression or classification model
- Determine the following beforehand
  - distance metric (ℓ_2 or ℓ_1 norms, etc)
  - number (k) of nearest neighbors
- k-NN is a non-parametric model

Advantage: Non-parametric.

Disadvantage: In the (vanilla version) of k-NN. need to compare Xnew w/ all existing data pts

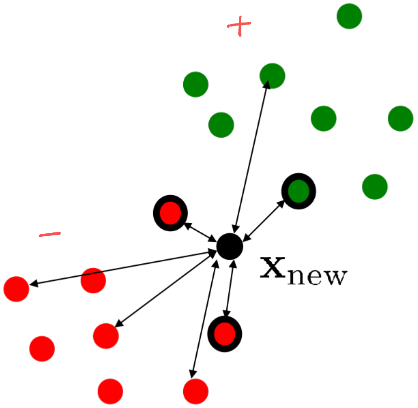
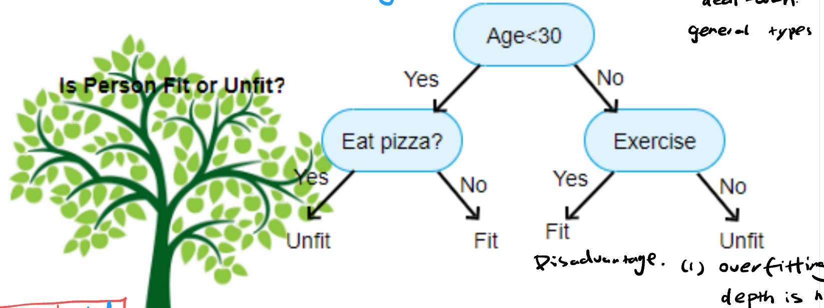


Figure: A toy example (k=3)

Review: Decision Tree

Idea: mimicking the process in human-based decision-making.

Advantage: Natural
deal with
general types of data



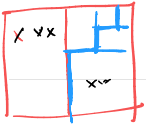
Disadvantage. (1) overfitting if depth is high

Person 1: age=20, eat pizza, exercise

(2). Variance can be high!

Person 2: age=40, no eat pizza, exercise

reduce it!



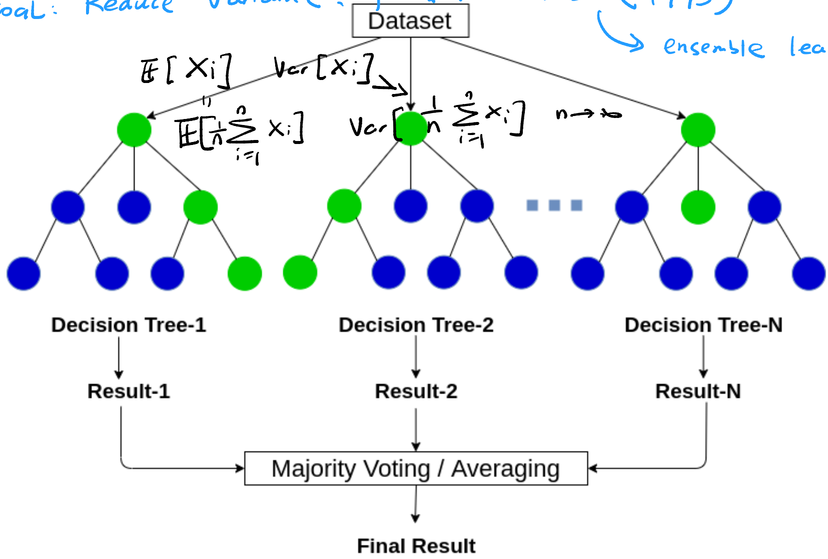
Q: Decison boundary Linear. ?
Non linear ? ✓

Review: Random Forest

Goal: Reduce variance of DT models (1995) → ensemble learning

$\mathbb{E}[X_i]$ $\quad$ $\mathrm{Var}[X_i] \rightarrow \mathrm{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right]$ $\quad$ $n \to \infty$

$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right]$



Review: Support Vector Machine

(1964.)

- Margin width:

$$M = (\mathbf{x}^+ - \mathbf{x}^-) \cdot \mathbf{n}$$
$$= (\mathbf{x}^+ - \mathbf{x}^-) \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$

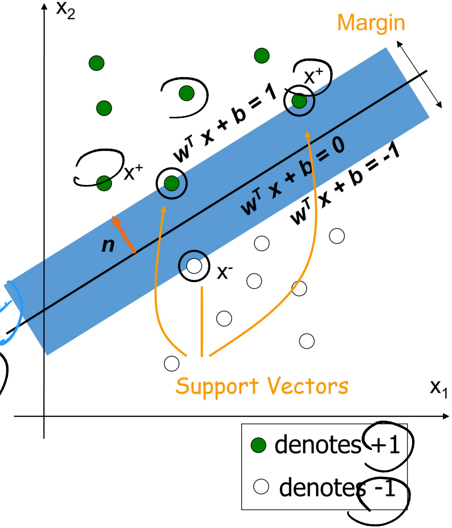
- Maximum margin classifier

(only works well when data is linearly separable)

$$\min_{\mathbf{w},\, b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{8}$$
$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i$$

Q: DB Linear ✓

Nonlinear ✗



Review: Support Vector Machine

If data is not linearly separable.

- Margin width:

$$M = (\mathbf{x}^+ - \mathbf{x}^-) \cdot \mathbf{n}$$

$$= (\mathbf{x}^+ - \mathbf{x}^-) \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$

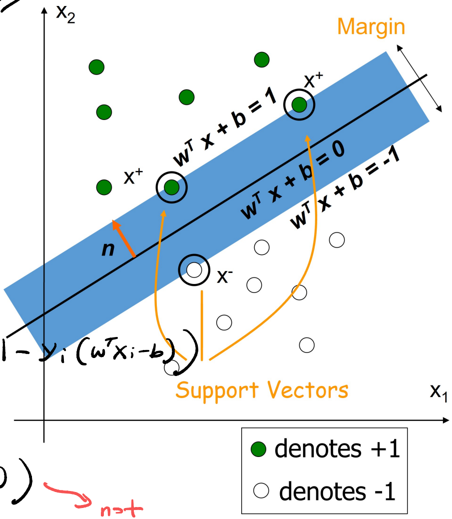
- Soft ~~Maximum~~ margin classifier

$$\min_{\mathbf{w},\, b} \; \cancel{\frac{1}{2}}\,\lambda \|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max\left(0,\, 1 - y_i(w^T x_i - b)\right) \qquad (8)$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \cancel{\geq} 1, \forall i$$

$$\geq 1 - \underbrace{\max\left(0,\, 1 - y_i(w^T x_i - b)\right)}_{\text{hinge loss}}$$

not differentiable



Review: Support Vector Machine

- Dual problem

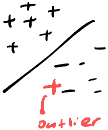
$$\max_{\alpha} \; \mathcal{L}_{\mathcal{D}}(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\top} \mathbf{x}_j \tag{9}$$

quadratic programming

$$\text{s.t.} \; \sum_{i=1}^{N} \alpha_i y_i = 0, \; \begin{cases} \alpha_i \le \frac{1}{2n\lambda} \\ \alpha_i \ge 0, i = 1, \ldots, N \end{cases}$$

other methods { • sub-gradient • coordinate descent.

- Kernel SVM (Linear $\xrightarrow{\text{generalize}}$ nonlinear)
  - replace $\mathbf{x}$ with $\phi(\mathbf{x})$
  - $\phi(\mathbf{x}_i)^{\top}\phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$
  - $k(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function, e.g, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$
- Slacked SVM

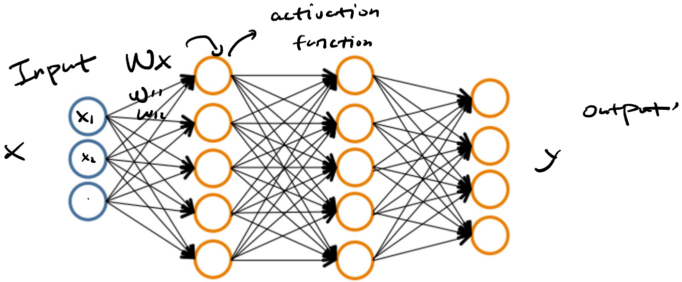


$$\min_{\mathbf{w}, b, \xi} \; \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i \tag{10}$$

$$\text{s.t.} \; y_i(\mathbf{w}^{\top}\mathbf{x}_i + b) \ge 1 - \xi_i, \; \xi_i \ge 0, \; i = 1, \ldots, N$$

Review: Neural Networks

- Fully connected feedforward network (multi-layer perceptron, MLP)



$$\mathbf{h}^{(1)} = f^{(1)}(\mathbf{x}) \quad \mathbf{h}^{(2)} = f^{(2)}(\mathbf{h}^{(1)}) \quad \ldots \quad \mathbf{y} = f^{(L)}(\mathbf{h}^{(L-1)})$$

Or $\quad \mathbf{y} = f^{(L)} \circ \cdots \circ f^{(1)}(\mathbf{x})$

Review: Neural Networks

- Convolutional neural network (CNN)

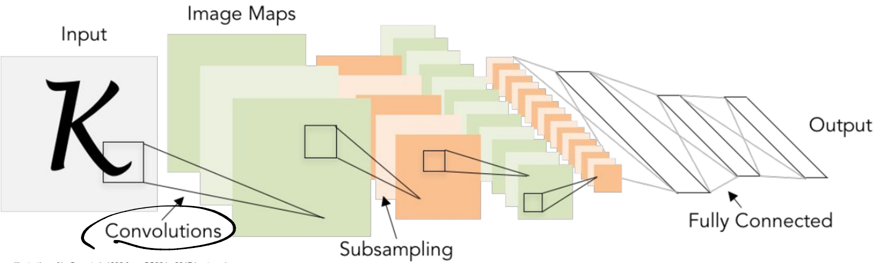


Illustration of LeCun et al. 1998 from CS231n 2017 Lecture 1

LeNet-5

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f(m)\, g(n-m)$$

Review: Neural Networks

- Recurrent neural network (RNN) (create a control system)

historical info.

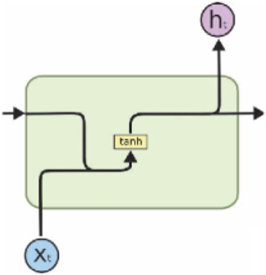
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

some function with parameters W



- works well for sequential data.

NLP

text

human voice

$X = (x_1, x_2, \dots, x_n)$

↓

letter

Review: Neural Networks

- Recurrent neural network (RNN)

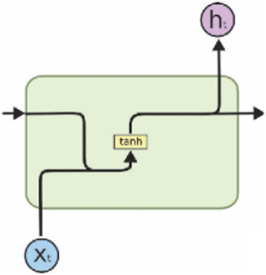
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

some function
with parameters W

old state

input vector at
some time step



- Other models for sequential data
  - LSTM, GRU, etc
  - Transformer
    - Widely used in LLMs such as GPTs
    - Not covered by DDA3020 and DDA4210

Review: Classification on Real Data

Classification on MNIST handwritten digits dataset

`http://yann.lecun.com/exdb/mnist/`

Figure: Samples of MNIST (28×28 gray-scale images, 60k for training, 10 k for testing)

classifier	test error rate (%)
linear classifier (least squares)	12.0
k-nearest-neighbors	5.0
generalized linear classifier (Gaussian basis 1000)	3.6
neural network (MLP) 500-300 HU, softmax	1.53
~~CNN LeNet-5~~	0.95
SVM (Gaussian kernel)	1.4

Review: Classification on Real Data

Classification on Fashion-MNIST dataset

https://cloudxlab.com/blog/fashion-mnist-using-machine-learning/

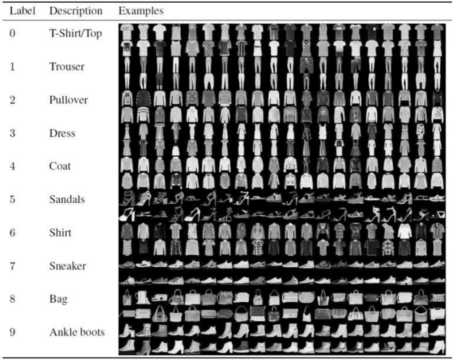


Figure: Samples of Fashion-MNIST (28×28 gray-scale images, 60k for training, 10 k for testing)

classifier	test error rate (%)
softmax	15.3
decision tree	21.06
random forest	15.18
neural network (MLP) (256-128-100 HU)	12.6
CNN	<8
HOG+SVM	7.4
Google AutoML	6.1

More results are at https://github.com/zalandoresearch/fashion-mnist

Review: K-Means Clustering

- Clustering (unsupervised learning): given a set of D-dimensional data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, partition them into K clusters such that each data point is similar to the data in the same cluster and dissimilar to the data in different clusters.

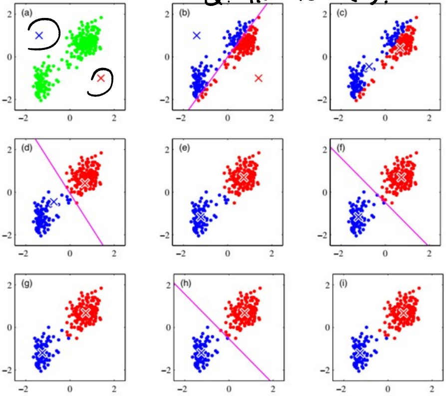
K=2

Q: How to decide K?
Q: How to initialize?
Q: How to d(·)?

- Denote cluster j by $\mathcal{C}_j$ and let $\boldsymbol{\mu}_j$ be the centroid of $\mathcal{C}_j$. K-means clustering minimizes

$$J(\mu) = \sum_{j=1}^{K} \sum_{\mathbf{x} \in \mathcal{C}_j} \|\mathbf{x} - \boldsymbol{\mu}_j\|^2 \quad (11)$$

- Algorithm (alternate)
  1. Assign each data point to the closest center
  2. Update the cluster center



Review: Gaussian Mixture Models

Idea: consider using conditional Gaussians to represent data distribution.

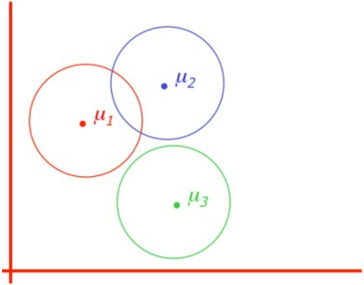
- Multivariate Gaussian distribution

$$p(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right)$$

- Gaussian mixture distribution

$$p(\mathbf{x}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)$$

  - K different Gaussian distributions
  - $\{\pi_j\}$: mixing coefficients
  - $\sum_{j=1}^{K} \pi_j = 1, \quad 0 \leq \pi_j \leq 1$



- Algorithm: Expectation-Maximization

Review: Principal Component Analysis

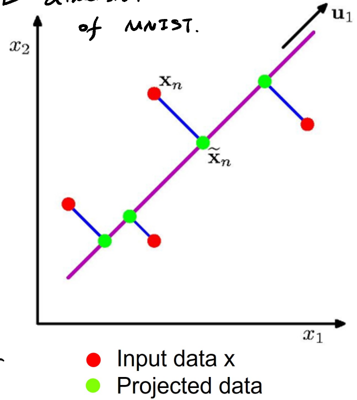
- PCA: find the orthogonal projection of data onto a lower-dimensional subspace that
  - maximizes the variance of projected data
  - or minimizes the reconstruction error, i.e.,

$$J = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_i - \mathbf{U}\mathbf{U}^\top \mathbf{x}_i\right\|^2 \quad (12)$$

SVD.

* $\boldsymbol{x} \in \mathbb{R}^D, \quad \boldsymbol{U} \in \mathbb{R}^{D\times d}$

- Doesn't work well for 2D dimension reduction of MNIST.



- Solution of PCA: eigenvalue decomposition or singular value decomposition

Review: More Topics (optional)

- Bayes' theorem, maximum likelihood estimation (MLE), maximum a posteriori estimation (MAP)
- Classification evaluation metrics
  - Precision, recall, accuracy, F1-score, AUC (TPR/FPR)
- Cross-validation
- Over-/under-fitting and bias-variance trade-off
- Expectation maximization
- Kernel density estimation
- Clustering evaluation metrics