THE UNIVERSITY *of* ADELAIDE

ISML_1: Overview of Machine Learning and Essential Mathematic Skills for Machine Learning

Lingqiao Liu
University of Adelaide

What's your impression about Machine Learning

Outlines

- Course Introduction
- What is machine learning and its application
- Machine Learning taxonomy and framework
- Mathematic basics in Machine Learning
  - Basic algorithmic calculations
  - Linear algebra: vector, matrix
  - Matrix calculus
  - Optimization
  - Probability theory

Outlines

- Course Introduction
- What is machine learning and its application
- Machine Learning taxonomy and framework
- Mathematic basics in Machine Learning
  - Basic algorithmic calculations
  - Linear algebra: vector, matrix
  - Matrix calculus
  - Optimization
  - Probability theory

Course content

- Focus on basic concepts and algorithms in machine learning, traditional and statistic machine learning technology
  - There are courses focusing on advanced topics, e.g., deep learning or application-oriented content, e.g., applied machine learning
  - This course is expected to lay a good foundation for your future study
  - It can be math intensive

Course content

Course Schedule (11 to 1 PM Wednesday)
(subject to minor change):

Week 1: Overview of Machine Learning, Mathematical basics for machine learning

Week 2: Basic concepts in Machine Learning and KNN classifiers

Week 3: Linear Classifier (Linear SVM)

Week 4: Regression

Week 5: Boosting and Random forest

Week 6: PCA, LDA and dimensionality reduction

Week 7: Unsupervised Learning: Clustering

Week 8: Kernel Method

Week 9: Deep Learning

Week 10: Semi-supervised learning and Unsupervised feature Learning

Week 11: Guest Lecture (TBA)

Week 12: Generative Model and Course Review

Course Introduction

- Course coordinator: Dr. Lingqiao Liu
- Colecturer: Dr. Dong Gong
  - Email: lingqiao.liu@adelaide.edu.au dong.gong@adelaide.edu.au
  - Office: 1.23 Australian Institute for Machine Learning
- Tutors:
  - Jinan Zou, Qiaoyang Luo, Bowen Zhang
- Components and assessments
  - 12 Lectures: 11 main lectures + 1 guest lecture
  - 4 workshops
  - 4 assignments (50%)
    - 1 simple assignment on solving several math problems (related to ML) (5%)
    - 3 assignments involves implementing machine learning algorithms (coding + report) (15% each)
  - Final exam (50%)
    - Hurdle 40%

Prerequisite

- Linear algebra
  - Vector, inner product, outer product, norm, Euclidean distance
  - Matrix, basic operations (addition, multiplication, inverse)
  - Determinant, trace, derivatives
  - Eigenvectors and eigenvalue
- Probability theory and Statistics
  - Random variable, probability density function
  - Mean, variance, covariance matrix
  - Statistical independence, conditional probability
  - Law of total probability, Bayes rule
  - Normal (Gaussian) distribution

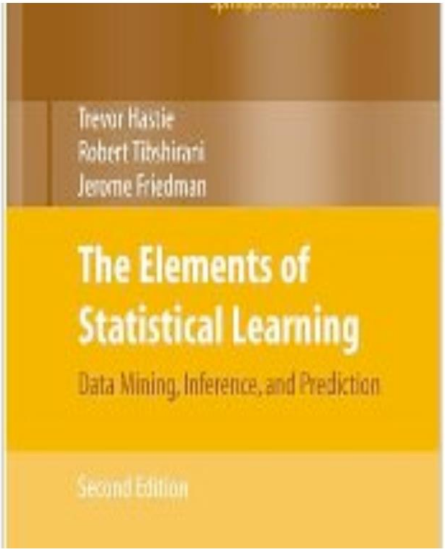
Prerequisite

- Programming skills:
  - Python (essential)
  - Matlab or other programming languages (optional)

Course delivery

- Face to face + online (tentative)
- Lectures will be live-streamed and recorded. Recording will be uploaded to MyUni (Echo360)
- Please check announcement and discussion forum regularly
  - I will check the discussion forum and answer your question every weekdays (at least once per day)
  - If you have urgent questions, please email me

References

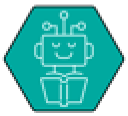


Downloadable from the authors' website. Just google "Trevor Hastie"

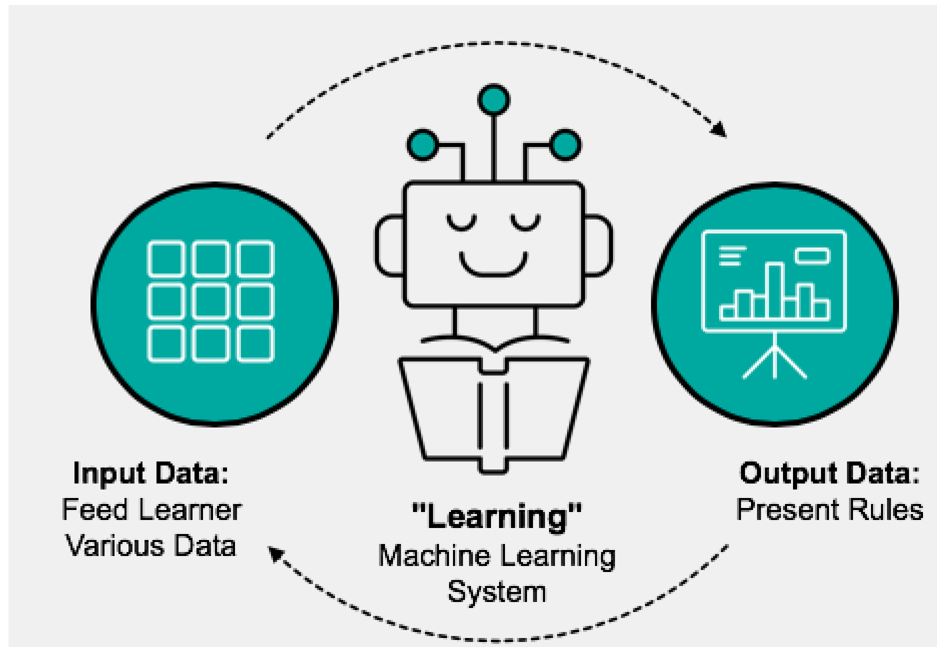
Outlines

- Course Introduction
- What is machine learning and its application
- Machine Learning taxonomy and framework
- Mathematic basics in Machine Learning
  - Basic algorithmic calculations
  - Linear algebra: vector, matrix
  - Matrix calculus
  - Optimization
  - Probability theory

What is Machine Learning?



- Learns autonomously through a dynamic feedback loop
- Increasingly self-healing, self-organizing, and self-architecting



What is Machine Learning?

- Data driven vs Expert Systems
  - Do not rely on the expert to specify the rules
  - Less expensive but more robust
  - Quick adapt to new environment

Applications

- Numerous applications
  - Image recognition, Speech recognition, Machine translation, recommendation systems
  - Fake image/audio/video generation, automatic music composition
  - Drug discovery, Computer-Aided Diagnosis, etc.
  - …
- Top 10 Applications of Machine Learning | Machine Learning Applications & Examples | Simplilearn – YouTube
- A new paradigm in science and engineering
  - Learning to predict
  - Learning to act
  - Learning to generate

Outlines

- Course Introduction
- What is machine learning and its application
- Types of Machine Learning systems
- Basic concepts in Machine Learning
- Mathematic basics in Machine Learning
  - Basic algorithmic calculations
  - Linear algebra: vector, matrix
  - Matrix calculus
  - Optimization
  - Probability theory

Types of Machine Learning systems

- Lots of categorization perspectives
  - From the availability of supervision
  - From the methodology
  - From the purpose of a machine learning system
  - …
- Availability of supervision
  - Three main categories:
    - Supervised learning
    - Unsupervised learning
    - Reinforcement learning
  - Other hybrid types: Semi-supervised learning and weakly supervised learning
- Types of the mapping function
  - Shallow machine learning
  - Deep machine learning

Supervised learning

- In supervised learning, the desired output is provided and the loss function measures the discrepancy between the output of mapping function and the true output
- Training dataset

Given a **training set** of N example **input-output pairs**

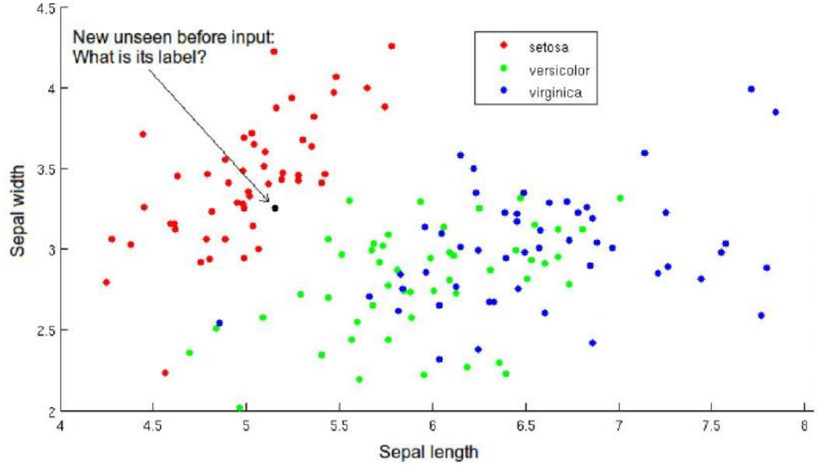
$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N),$$

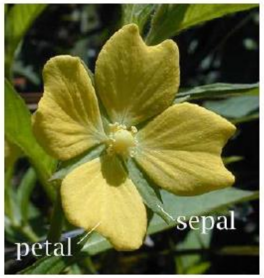
- Loss function

$$\sum_i \mathcal{L}(f(x_i), y_i)$$

Example

Example: Given measurements of sepal length and sepal width, identify the **type** of flower.



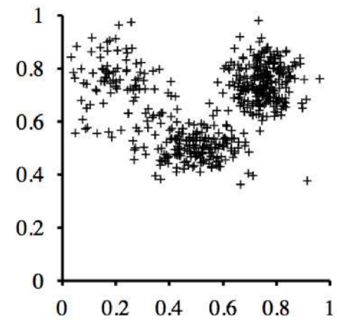


Here each input value $\mathbf{x}_i$ is a two-dimensional vector, while each target value y_i can be setosa, versicolor or virginica.

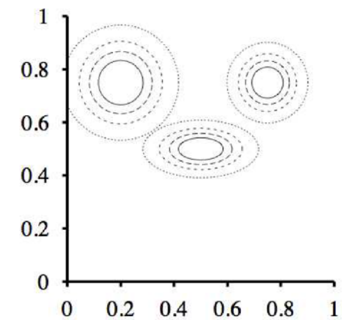
Unsupervised learning

- Learning patterns when no specific target output values are supplied
- Examples:
  - Clustering: group data into groups
  - Building probabilistic model to explain data
  - Anomaly detection

Example: Finding clusters in 2D input data.



(a) Input data with unknown cluster memberships.

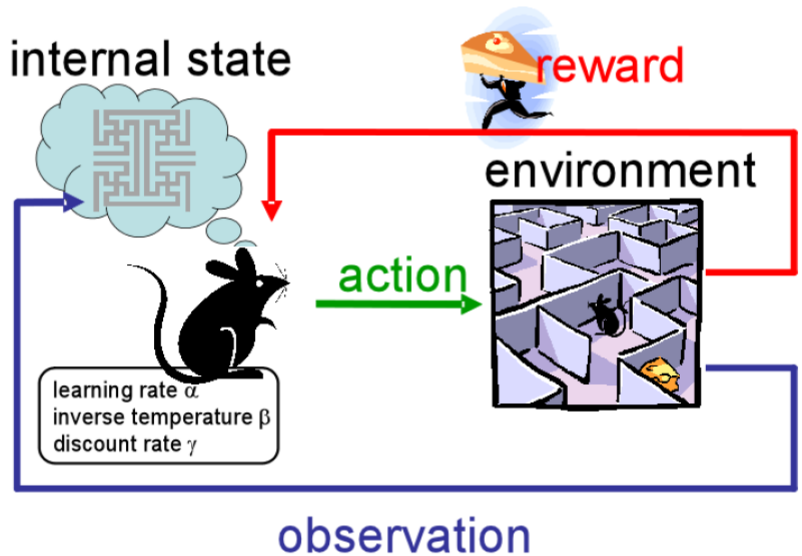


(b) A hypothesis clustering.

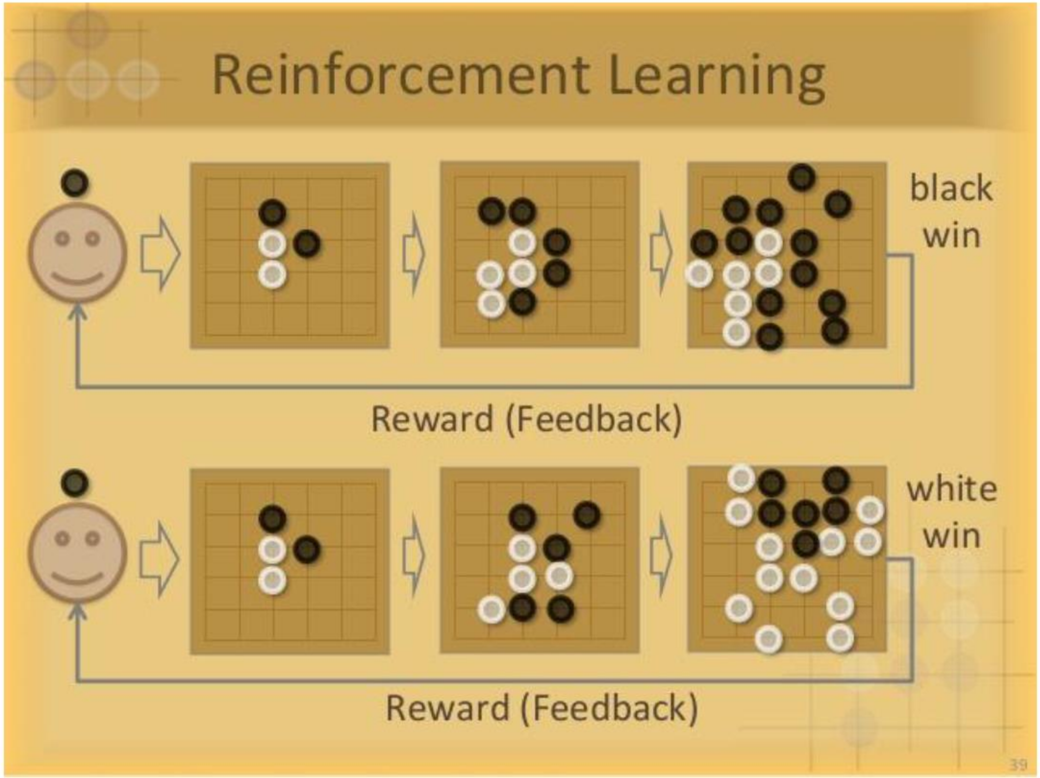
Reinforcement learning

Learning what actions to take in order to maximise some **reward** or **utility**.

The reward (or penalty) is given after a **sequence of actions**. This differs from supervised learning in that explicit input-output examples are not available.



Reinforcement learning



Types of machine learning: shallow vs. deep

- Traditional machine learning
  - Important step: feature design
  - Usually work with "feature vectors"
  - Mapping function is simple, with relatively small number of parameters
  - Works well if the input can be captured by vectors, small to medium number of samples
- Deep learning
  - Allows raw input
  - End-to-end learning
  - Complex models, with millions of parameters
  - Works well if the "right feature" is unknown or the input is complex and a large number of samples are available

Outlines

- Course Introduction
- What is machine learning and its application
- Machine Learning taxonomy and framework
- Mathematatic basics in Machine Learning
  - Basic algorithmic calculations
  - Linear algebra: vector, matrix
  - Matrix calculus
  - Optimization
  - Probability theory

The workflow of machine learning systems

- Problem formulation
  - What is the input? What is the expected outcome
- Data collection
  - Collect data
  - Annotation
- Design machine learning algorithm
  - Choose the machine learning model
  - Choose the objective function
- Training machine learning model: Learn the decision system from training data
- Applying machine learning model

Element of machine learning systems

- Input/Output
  - Input: can be feature vectors, text, images, videos, symbolic sequences
  - Output: class label, continues value, structured output or a sequences of actions
- Mapping function
  - Map input to the desirable output
  - Many possible mappings, e.g., same form but different parameters
- Loss function
  - Judge if the mapping function is good enough

Element of machine learning systems

- Machine learning is a process of finding the optimal mapping function

$$F^* = argmin_F \, \mathcal{L}(F(X))$$

Outlines

- Course Introduction
- What is machine learning and its application
- Machine Learning taxonomy and framework
- Mathematic basics in Machine Learning
  - Basic algorithmic calculations
  - Linear algebra: vector, matrix
  - Matrix calculus
  - Optimization
  - Probability theory

Summation and Product

- Commonly used operations in Statistic Machine Learning
- Summation notations
  - Summation

$$\sum_{i=1}^{N} a_i = \sum_i a_i = \sum_j a_j$$

  - Summation with two indices

$$\sum_{i=1}^{M} \sum_{j=1}^{N} a_{ij} = \sum_{i,j} a_{ij}$$

Summation and Product

- A useful formula (a little bit counter-intuitive)

$$\sum_{i=1}^{M} \sum_{j=1}^{N} a_i b_j = \left(\sum_i a_i\right) \left(\sum_j b_j\right) = \left(\sum_i a_i\right) \left(\sum_i b_i\right)$$

Summation and Product

- A useful formula (a little bit counter-intuitive)

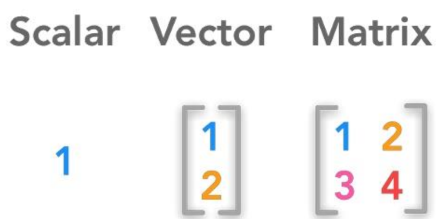
$$\sum_{i=1}^{M} \sum_{j=1}^{N} a_i b_j = \left(\sum_i a_i\right) \left(\sum_j b_j\right) = \left(\sum_i a_i\right) \left(\sum_i b_i\right)$$

- Proof

$$\sum_{i=1}^{M} \sum_{j=1}^{N} a_i b_j = \sum_i a_i \left(\sum_j b_j\right) = \left(\sum_i b_i\right) \left(\sum_i a_i\right)$$

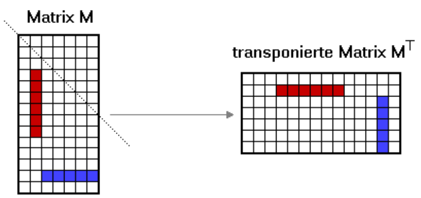
Linear algebra: vector, matrix and basic matrix operations

- Vectors and matrix



- Basic operations
  - Multiplication

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ c_7 & c_8 & c_9 \end{bmatrix}$$



  - Transpose

$$\begin{bmatrix} 3 & 3 \\ 5 & -3 \\ 1 & 3 \\ 0 & -2 \\ 5 & -2 \end{bmatrix}^T = \begin{bmatrix} 3 & 5 & 1 & 0 & 5 \\ 3 & -3 & 3 & -2 & -2 \end{bmatrix}$$

  - Inverse

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

Matrix multiplication

- View matrix as a set of vectors

$$\mathbf{A}\mathbf{b} = \underbrace{[\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n]}_{\text{Row vectors}}^T \mathbf{b} = [\mathbf{a}_1^\top b, \mathbf{a}_2^\top b, \cdots, \mathbf{a}_n^\top b]^\top$$

$$\mathbf{A}\mathbf{\Lambda} = \underbrace{[\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n]}_{\text{Column vectors}} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} = [\lambda_1 \mathbf{a}_1, \lambda_2 \mathbf{a}_2, \cdots, \lambda_n \mathbf{a}_n]$$

Inner product and norms

- Inner product between two vectors

$$< \mathbf{x}, \mathbf{y} >= \mathbf{x}^T \mathbf{y} = \sum_i x_i y_i$$

- Vector Norms
  - Measure the length of the vector
  - Not unique: could have infinite number of definitions
  - Commonly used ones

$$l_2 \text{ norm: } \|x\|_2 = \sqrt{\sum_i x_i^2} \qquad l_1 \text{ norm: } \|x\|_1 = \sum_i |x_i|$$

$$l_p \text{ norm: } \|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$$

Trace and Matrix Norm

- Definition $Tr(A) = \sum_i a_{ii} \quad Tr(a) = a$
- Properties

$$Tr(X^\top Y) = Tr(XY^\top) = Tr(Y^\top X) = Tr(YX^\top)$$

$$Tr(A) + Tr(B) = Tr(A + B)$$

- Frobenius norm

$$\|A\|_F = \sqrt{\sum_{ij} a_{ij}^2}$$

- Relationship to Trace

$$\|A\|_F = \sqrt{\sum_{ij} a_{ij}^2} = \sqrt{Tr(AA^\top)} = \sqrt{Tr(A^\top A)}$$

Linear Subspace

- For k vectors $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$, all of their linear combinations form a linear space, i.e.,

$$\{\mathbf{x} | \mathbf{x} = t_1\mathbf{v}_1 + t_2\mathbf{v}_2 + \cdots + t_k\mathbf{v}_k\}$$

- Basis: if $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$ are orthogonal to each other
  - Equivalent to the coordinate in a space

$$< \mathbf{v}_i, \mathbf{v}_j >= 0 \quad \forall \quad i \neq j$$

Eigen vector and eigen values

- Eigenvalue and Eigenvectors

$$Au = \lambda u$$

- Eigen vectors is not unique
  - Apply scaling and addition operations will also produce eigenvectors
  - So the eigenvectors corresponding to an eigenvalue form a linear subspace
  - Usually we only interested in a set of independent eigenvectors, each one will correspond to an eigenvalue
  - Modern solver will return a set of eigenvalues and their corresponding vectors

Matrix decomposition

- Matrix can be decomposed into the combination (usually product) of special matrices
- Eigen decomposition

$$A = Q\Lambda Q^{-1}$$

where Λ is a diagonal matrix, with its i-th diagonal value be the i-th eigenvalue of A. Q is a matrix with its ith column be the eigenvector corresponding to ith eigenvalue.

  – When A is symmetric, i.e. $A = A^\top$

$$A = Q\Lambda Q^\top \qquad Q^\top Q = QQ^\top = I$$

- Related topic: Singular value decomposition

Optimization

- Optimization: find a variable that can gives the minimal (maximal) value of the objective function
  - The variable may under certain constrains, say, $x \in \Omega$
  - Ω is called the feasible set of x
- In machine learning, we are going to learn a mapping function $f(x; \lambda)$
- We will have a loss function or objective function to measure its performance

$$\mathcal{L}(\lambda) = \mathcal{J}(f(x; \lambda))$$

Optimization problem

- General form

$$\min_{x \in \Omega} \mathcal{L}(x)$$

- Example

$$\min_{x} \mathcal{L}(x)$$
$$s.t. \quad g(x) \leq 0$$

- Could be simple or very difficult, depend on the type of objective function and the type of constrains

Equivalence of Optimization problem

- In optimization, we often convert an optimization problem to another equivalent optimization problem.
  - Consider Op1 and Op2, if we know the solution of Op2, we can know Op1, they can be deemed equivalent.
- Example

$$\max_x -x^2 + x$$

$$\Downarrow$$

$$\max_x -(x^2 - 2\tfrac{1}{2}x + \tfrac{1}{4}) + \tfrac{1}{4} = -(x - 1/2)^2 + 1/4$$

$$\Downarrow$$

$$\min_x (x - 1/2)^2$$

More examples

$$\min_{\{x_i\}} \sum_i f(x_i) + \sum_i |x_i|$$

$$\Downarrow$$

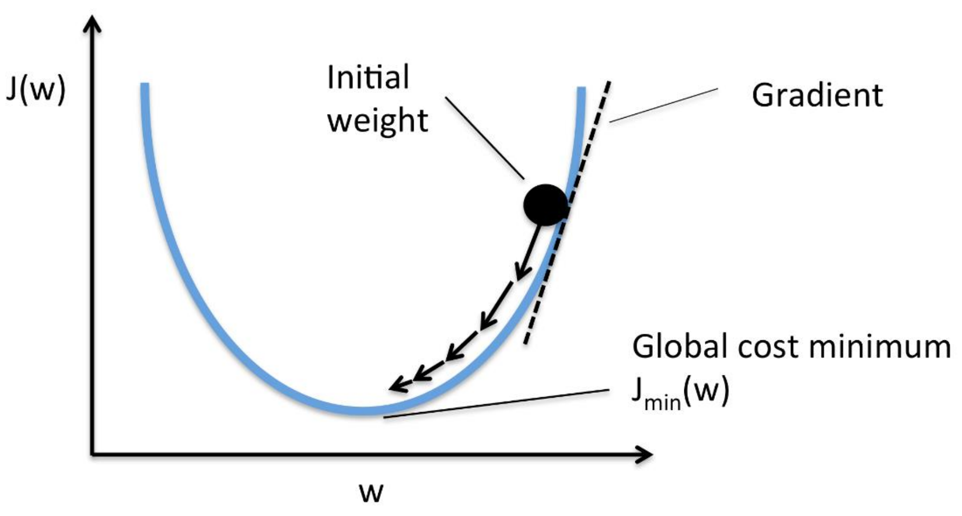
$$\min_{\{x_i,\xi_i\}} \sum_i f(x_i) + \sum_i \xi_i$$

$$s.t. \quad x_i \leq \xi_i$$

$$-x_i \leq \xi_i \quad \forall x_i, \xi_i$$

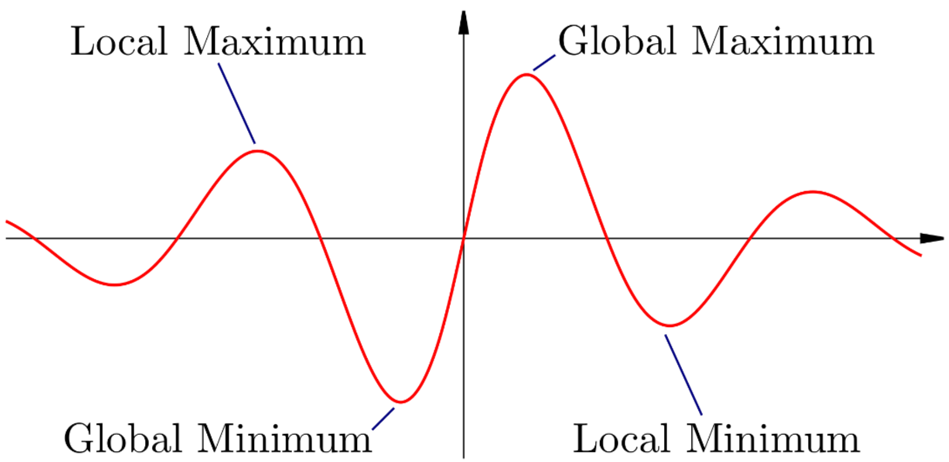
Solution to optimization problems

- General Purposed Solution
  - Zero-order method
  - Frist-order method
  - Second-order method



Solution to optimization problems

- Global Minimum and Local Minimum



- At Local minimum, gradient equals 0.
  - If we know an **unconstrained** optimization has local minimum = Global Minimum, we can solve $\frac{\partial f(x)}{\partial x} = 0$ to find the optimal solution

Type of optimization problems

- Many of them
  - Continuous vs. Discrete: binary or Integer variables
  - Linear vs. Nonlinear
  - Convex vs. nonconvex

- Convex optimization problem
  - Global optimum = Local optimum

Matrix calculus

- For functions that involve matrices or vectors
  - Case 1: Vector/Matrix variable and scalar output
  - Case 2: Vector/Matrix variable and vector output
- Definition
- Application
  - Similar

$$\frac{\partial y}{\partial \mathbf{x}} \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix}.$$

Matrix calculus

- Properties
- More info
  - [Matrix Calculus]()
- Trick to memorize
  - Analogy to scalar case
  - Check dimensions

Identities: vector-by-vector $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$

Condition	Expression	Numerator layout, i.e. by $\mathbf{y}$ and $\mathbf{x}^\mathsf{T}$	Denominator layout, i.e. by $\mathbf{y}^\mathsf{T}$ and $\mathbf{x}$
a is not a function of **x**	$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} =$	$\mathbf{0}$	
	$\frac{\partial \mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{I}$	
A is not a function of **x**	$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{A}$	$\mathbf{A}^\top$
A is not a function of **x**	$\frac{\partial \mathbf{x}^\top \mathbf{A}}{\partial \mathbf{x}} =$	$\mathbf{A}^\top$	$\mathbf{A}$
a is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	
v = *v*(**x**), **u** = **u**(**x**)	$\frac{\partial v\mathbf{u}}{\partial \mathbf{x}} =$	$v\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{u}\frac{\partial v}{\partial \mathbf{x}}$	$v\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}\mathbf{u}^\top$
A is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial \mathbf{A}\mathbf{u}}{\partial \mathbf{x}} =$	$\mathbf{A}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{A}^\top$
u = **u**(**x**), **v** = **v**(**x**)	$\frac{\partial (\mathbf{u}+\mathbf{v})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	
u = **u**(**x**)	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}$
u = **u**(**x**)	$\frac{\partial \mathbf{f}(\mathbf{g}(\mathbf{u}))}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}$

Matrix calculus

- More information
- Exercise

$$\min_{\mathbf{x}} \|Ax - b\|_2^2$$

- Hint

$$\|Ax - b\|_2^2 = (Ax - b)^\top (Ax - b)$$

Matrix calculus

- More information
- Exercise

$$\min_{\mathbf{x}} \|Ax - b\|_2^2$$

- Hint

$$\|Ax - b\|_2^2 = (Ax - b)^\top (Ax - b)$$

$$\frac{\partial \|Ax-b\|_2^2}{\partial x} = 2A^\top (Ax - b) = 2A^\top Ax - 2A^\top b$$

$$Solve \quad \frac{\partial \|Ax-b\|_2^2}{\partial x} = 2A^\top Ax - 2A^\top b = 0$$

$$x = (A^\top A)^{-1} A^\top b$$

Probability and random variable

- Random variable: a way describe the random experiment outcome
- Probability distribution
  - For discrete random variable, its probabilistic distribution is characterised by Probability Mass Function

$$p_X(x_i) = P(X = x_i)$$

$$\sum p_X(x_i) = 1$$
$$p(x_i) > 0$$
$$p(x) = 0 \text{ for all other x}$$

  - For continuous random variable, the counterpart of Probability Mass Function is probability density function (PDF)

$$P(a \leq X \leq b) = \int_a^b f(x)dx$$

Probability and statistics

- Commonly used PDF
  - Uniform distribution

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \le x \le b, \\ 0 & \text{for } x < a \text{ or } x > b \end{cases}$$

  - Gaussian distribution

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

  - Multivariate Gaussian distribution

$$f(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu, \Sigma) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)}{\sqrt{(2\pi)^k|\Sigma|}}$$

More than one random variables

- Distribution of a collection of random variables
  - Consider the case of two random variables $f_{XY}(x, y)$ or $p(x, y)$
- Marginal distribution

$$p(x) = \int p(x, y)dy$$

- Conditional distribution

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(x,y)}{\int p(x,y)dx}$$

- Independence

$$p(x, y) = p(x)p(y) \qquad p(x|y) = p(x) \qquad p(y|x) = p(y)$$

More than one random variables

- Conditional independence

$$p(x, y|z) = p(x|z)p(y|z)$$

  – Note: conditional independence and independence are two different concepts

- Bayes rule

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dx}$$

Latent variable

- Sometimes, it is convenient to introduce an additional random variable and model the joint distribution

$$p(X, Z) = p(X)p(X|Z)$$

- Then the distribution over X can be calculated via marginalization

$$p(X) = \sum_z p(Z)p(X|Z)$$

- Usually introducing Z is necessary if we know the generative process of X (How X is sampled)

Example

- Imagine we have 3 biased dices; the outcome of each dice will be a random variable X_1, X_2, X_3 with distribution p_1, p_2, p_3
- We add another layer of randomness by choosing the dice randomly from a given distribution
- The final outcome will be a random variable Y

Example

- Imagine we have 3 biased dices; the outcome of each dice will be a random variable X_1, X_2, X_3 with distribution p_1, p_2, p_3
- We add another layer of randomness by choosing the dice randomly from a given distribution
- The final outcome will be a random variable Y
- We can define the choice made in dice selection as an additional random variable Z

$$p(Y, Z) = p(Y|Z)p(Z)$$

Expectations and Variance

- Discrete case

$$\mathrm{E}[X] = \sum_i x_i p_i$$

- Continuous case

$$\mathrm{E}[X] = \int_{\mathbb{R}} x f(x)\, dx.$$

- Variance

$$\begin{aligned}
\operatorname{Var}(X) &= \mathrm{E}\left[(X - \mathrm{E}[X])^2\right] \\
&= \mathrm{E}\left[X^2 - 2X\,\mathrm{E}[X] + \mathrm{E}[X]^2\right] \\
&= \mathrm{E}\left[X^2\right] - 2\,\mathrm{E}[X]\,\mathrm{E}[X] + \mathrm{E}[X]^2 \\
&= \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2
\end{aligned}$$