Math Background (II)

杜俊

jundu@ustc.edu.cn

Math Review

- Probability & Statistics
  - Bayes’ theorem
  - Random variables: discrete vs. continuous
  - Probability distribution: PDF and CDF
  - Statistics: mean, variance, moment
  - Parameter estimation: MLE

- Information Theory
  - Entropy, mutual information, information channel, KL divergence

- Function Optimization
  - Constrained/unconstrained optimization

- Linear Algebra
  - Matrix manipulation

Information Theory

- Claude E. Shannon (1916-2001, from Bell Labs to MIT): Father of Information Theory, Modern Communication Theory …
  - C. E. Shannon, "A Mathematical Theory of Communication", Parts 1 & 2, *Bell System Technical Journal*, 1948.

- Information of an event

$$I(A) = \log_2 1/\Pr(A) = -\log_2 \Pr(A)$$

- Entropy(Self-Information) : in bit/nat, amount of info in a R.V.
  - Entropy represents average amount of information in a R.V., the average uncertainty related to a R.V.

$$H(X) = -\sum_{x} p(x)\log_2 p(x) = \mathrm{E}[\log_2 \frac{1}{p(X)}]$$

Joint and Conditional Entropy

- Joint entropy: average amount of information (uncertainty) about two R.V.s.

$$H(X,Y)=\mathrm{E}[\log_2\frac{1}{p(X,Y)}]=-\sum_{x}\sum_{\mathrm{y}}p(x,y)\log_2 p(x,y)$$

- Conditional entropy: average amount of information (uncertainty) of Y after X is known.

$$H(Y\,|\,X)=\mathrm{E}[\log_2\frac{1}{p(Y\,|\,X)}]=-\sum_{x}\sum_{\mathrm{y}}p(x,y)\log_2 p(y\,|\,x)$$

- Chain rule for entropy

$$H(X_1,X_2,\ldots,X_n)=H(X_1)+H(X_2\,|\,X_1)+\cdots+H(X_n\,|\,X_1,\ldots,X_{n-1})$$

- Independence

$$H(X,Y)=H(X)+H(Y)\quad \text{or}\quad H(Y\,|\,X)=H(Y)$$

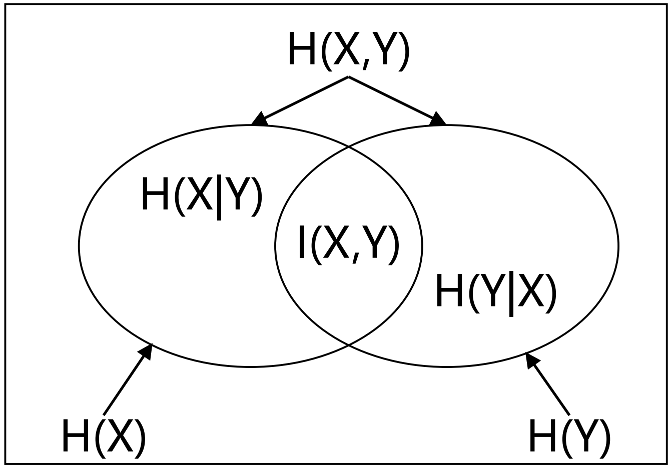
Mutual Information

- Average information about Y (or X) we can get from X (or Y).

$$I(X,Y)=H(X)-H(X\,|\,Y)=H(Y)-H(Y\,|\,X)=H(X)+H(Y)-H(X,Y)$$

$$I(X,Y)=\sum_{x\in X}\sum_{y\in Y}p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}$$

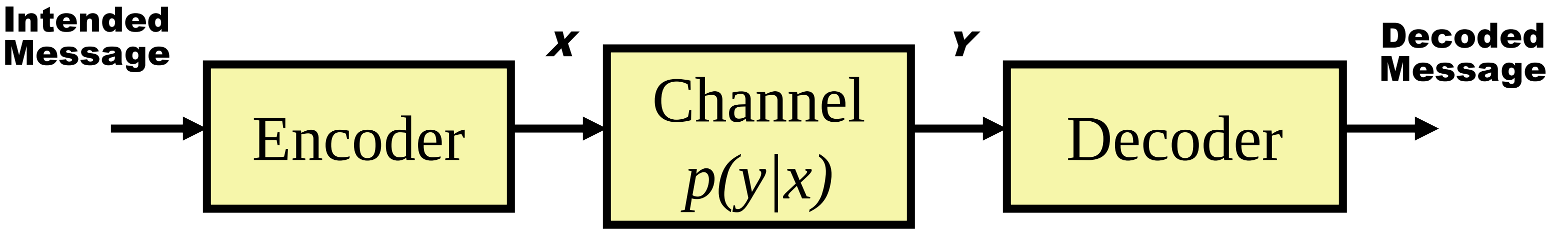
$$\text{or}\ \int_x\int_y p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}\text{dxdy}$$



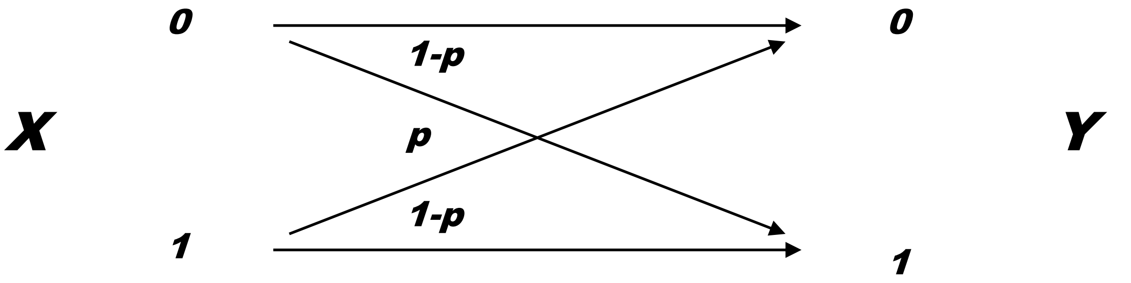
- Maximizing I(X,Y) is equivalent to establishing a closer relationship between X and Y, i.e., obtaining a low-noise information channel between X and Y.

Shannon’ s Noisy Channel Model

- Shannon’ s noisy channel model



- A binary symmetric noisy channel



$$I(X,Y)$$

- Channel capacity: the tightest upper bound on the rate of information that can be reliably transmitted over a communication channel

$$C = \max_{p(X)} I(X,Y) = \max_{p(X)}[H(Y) - H(Y|X)] = 1 - H(p) \leq 1$$

Example

- X,Y is equiprobable: Pr(X=0)=Pr(X=1)= Pr(Y=0)=Pr(Y=1)= 0.5

- p=0 (noiseless)

$$C = 1 - H(0) = 1$$

- p=0.1(weak noise)

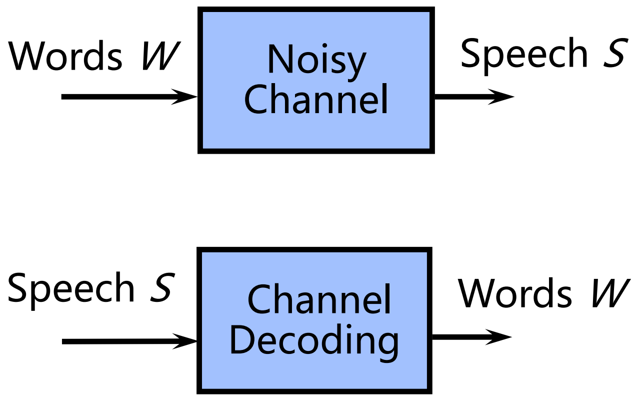
$$C = 1 - H(0.1) = 0.533$$

- p=0.4(strong noise)

$$C = 1 - H(0.4) = 0.03$$

Channel Modeling and Decoding

Speech Recognition



Words W

Noisy Channel

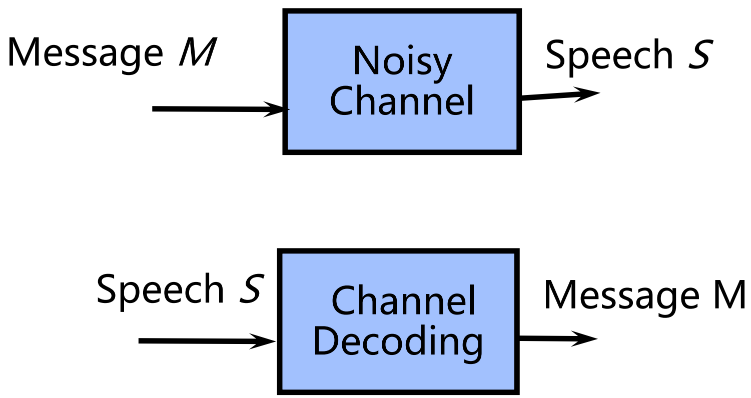
Speech S

Speech S

Channel Decoding

Words W

Speech Understanding



Message M

Noisy Channel

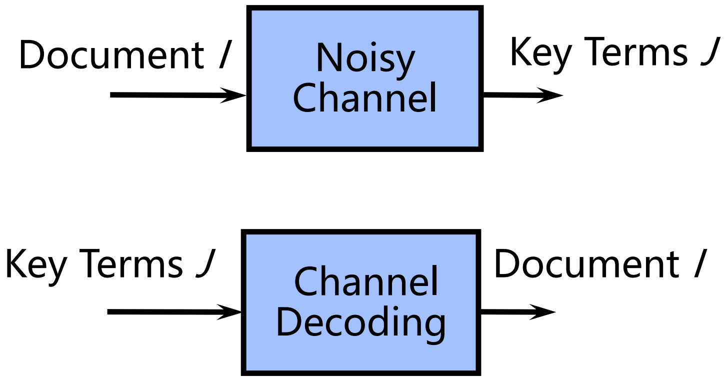
Speech S

Speech S

Channel Decoding

Message M

Information Retrieval



Document I

Noisy Channel

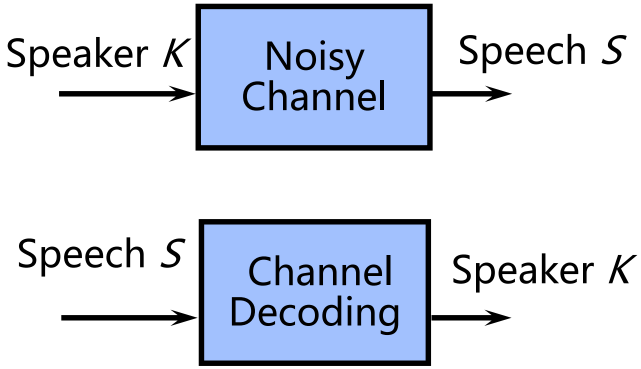
Key Terms J

Key Terms J

Channel Decoding

Document I

Speaker Identification



Speaker K

Noisy Channel

Speech S

Speech S

Channel Decoding

Speaker K

Bayes’ Theorem Application

- Bayes’ theorem for channel decoding

$$I^{*} = \arg\max_{I} P(I \mid \hat{O}) = \arg\max_{I} \frac{P(\hat{O} \mid I)P(I)}{P(\hat{O})} = \arg\max_{I} P(\hat{O} \mid I)P(I)$$

Application	**Input**	**Output**	***p(I)***	***p(O|I)***
Speech Recognition	**Word Sequence**	**Speech Features**	**Language Model (LM)**	**Acoustic Model**
Character Recognition	**Actual Letters**	**Letter images**	**Letter LM**	**OCR Error Model**
Machine Translation	**Source Sentence**	**Target Sentence**	**Source LM**	**Translation (Alignment) Model**
Text Understanding	**Semantic Concept**	**Word Sequence**	**Concept LM**	**Semantic Model**
Part-of-Speech Tagging	**POS Tag Sequence**	**Word Sequence**	**POS Tag LM**	**Tagging Model**

Kullback-Leibler Divergence

- KL divergence (Information divergence)
  - Introduced by Solomon Kullback and Richard Leibler in 1951
  - Distance measure between two PMFs or PDFs (relative entropy)
  - D(p||q) $\geq$0, D(p||q)=0 if and only if q=p, D(p||q)$\neq$ D(q||p)

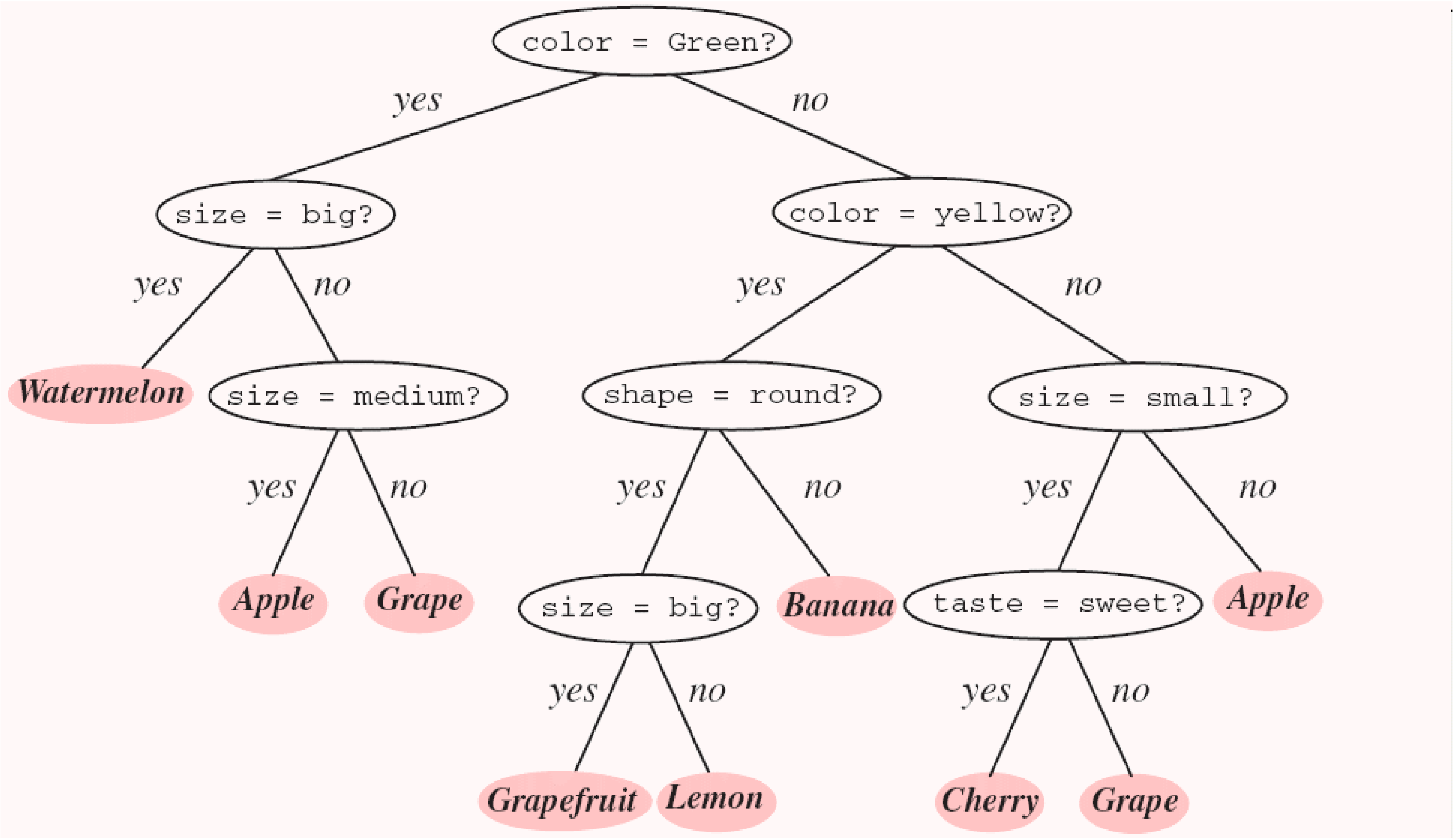
$$D(p\,||\,q) = \mathrm{E}_p[\log_2 \frac{p(x)}{q(x)}] = \sum_{x\in \mathrm{X}} p(x)\log_2 \frac{p(x)}{q(x)} \ \text{ or } \int_x p(x)\log_2 \frac{p(x)}{q(x)}\mathrm{d}x$$

- Mutual information is a measure of independence

$$I(X,Y) = D(p(x,y)\,||\,p(x)p(y))$$

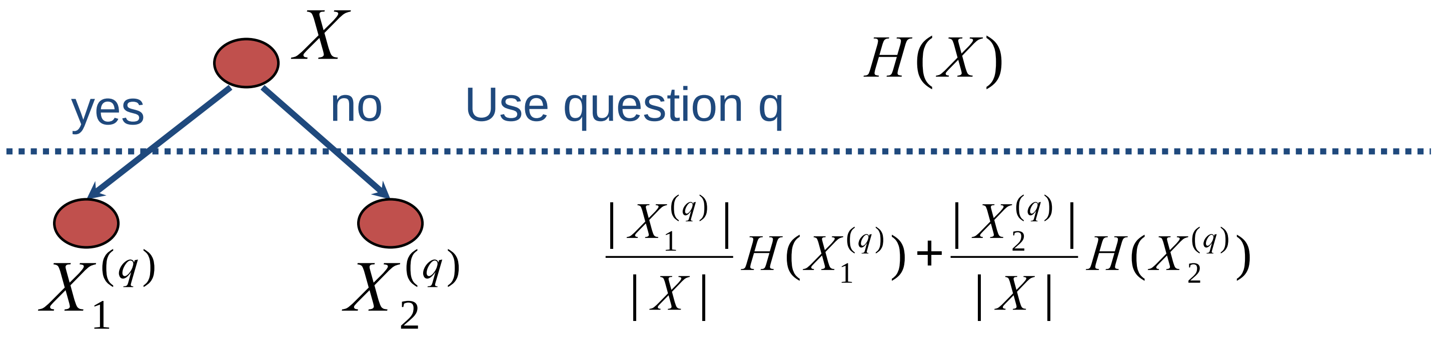
Classification: Decision Trees

Example: fruits classification based on features



Classification and Regression Tree (CART)

- Binary tree for classification
  - Each node is attached a YES/NO question
  - Traverse the tree based on the answers to questions
  - Each leaf node represents a class
- CART: Automatically grow a classification tree on a data-driven basis
  - Prepare a finite set of all possible questions
  - For each node, choose the best question to split the node
  - Maximum entropy reduction
  - Small entropy → more homogeneous the data is



$$H(X)$$

$$\frac{|X_1^{(q)}|}{|X|}H(X_1^{(q)})+\frac{|X_2^{(q)}|}{|X|}H(X_2^{(q)})$$

The CART algorithm

1) Question set: create a set of all possible YES/NO questions.
2) Initialization: initialize a tree with only one node which consists of all available training samples.
3) Splitting nodes: for each node in the tree, find the best splitting question which gives the greatest entropy reduction.
4) Go to step 3) to recursively split all its children nodes unless it meets certain stop criterion, e.g., entropy reduction is below a pre-set threshold OR data in the node is already too little.

CART method is widely used in machine learning and data mining:

1. Handle categorical data in data mining;
2. Acoustic modeling (allophone modeling) in speech recognition;
3. Letter-to-sound conversion;
4. Automatic rule generation
5. etc.

Optimization of objective function (I)

- Optimization:
  - Set up an objective function Q;
  - Maximize or minimize the objective function
- Maximization (minimization) of a function:
  - Differential calculus;
    - Unconstrained maximization/minimization

$$Q = f(x) \text{Þ} \frac{\mathrm{d} f(x)}{\mathrm{d}x} = 0 \text{Þ} \quad x = ?$$

$$Q = f(x_1, x_2, \cdots, x_N) \text{Þ} \frac{\text{¶} f(x_1, x_2, \cdots, x_N)}{\text{¶} x_i} = 0 \text{Þ} \quad ??$$

  - Lagrange optimization:
    - Constrained maximization/minimization

$$Q = f(x_1, x_2, \cdots, x_N) \quad \text{with constraint} \quad \mathrm{g}(x_1, x_2, \cdots, x_N) = 0$$

$$Q' = f(x_1, x_2, \cdots, x_N) + / \times g(x_1, x_2, \cdots, x_N)$$

$$\frac{\text{¶} Q'}{\text{¶} x_1} = 0, \frac{\text{¶} Q'}{\text{¶} x_2} = 0, \cdots, \frac{\text{¶} Q'}{\text{¶} x_N} = 0, \frac{\text{¶} Q'}{\text{¶} /} = 0$$

Karush–Kuhn–Tucker (KKT) Conditions

- Primary Problem:

$$\min_{\mathrm{x}} \quad f(\mathrm{x})$$

$$\text{subject to}$$

$$\mathrm{g}_i(\mathrm{x}) \pounds 0 \qquad (i = 1, \cdots, m)$$

$$h_j(\mathrm{x}) = 0 \qquad (j = 1, \cdots n)$$

- Introduce KKT multipliers:
  - For each inequality constraint: $m_i \quad (i = 1, \cdots, m)$
  - For each equality constraint: $/_j \quad (j = 1, \cdots, n)$

Karush–Kuhn–Tucker (KKT) Conditions

- Dual problem:
  - if x* is local optimum of the primary problem, x* satisfies:

$$\nabla f(\mathrm{x}^*) + \sum_{i=1}^{m} m_i \nabla g_i(\mathrm{x}^*) + \sum_{j=1}^{n} l_j \nabla h_j(\mathrm{x}^*) = 0$$

$$m_i \geq 0 \quad (i = 1, \cdots, m)$$

$$m_i g_i(\mathrm{x}^*) = 0 \quad (i = 1, \cdots, m)$$

- The primary problem can be alternatively solved by the above equations.

Optimization of objective function (II)

- Gradient descent (ascent) method:

$$Q = f(x_1, x_2, \cdots, x_N)$$

For any x_i, start from any initial value $x_i^{(0)}$

$$x_i^{(n+1)} = x_i^{(n)} \pm e \cdot \nabla_{x_i} f(x_1, x_2, \cdots, x_N)\,|_{x_i = x_i^{(n)}}$$

where $\nabla_{x_i} f(x_1, x_2, \cdots, x_N) = \dfrac{\partial f(x_1, x_2, \cdots, x_N)}{\partial x_i}$

  - Step size is hard to determine
  - Slow convergence
  - Stochastic gradient descent (SGD)

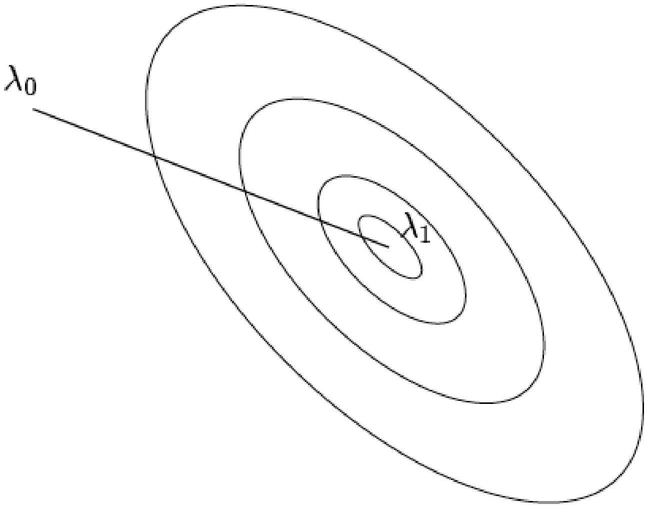
Optimization of objective function (II)

- Newton’ s method

$$Q = f(\mathrm{x})$$

Given any initial value x_0

$$f(\mathrm{x}) \approx f(\mathrm{x}_0) + \nabla f(\mathrm{x}_0)(\mathrm{x}-\mathrm{x}_0)^t + \frac{1}{2}(\mathrm{x}-\mathrm{x}_0)^t H(\mathrm{x}-\mathrm{x}_0)$$



$$H = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_N} \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_N} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_N} & \cdots & \frac{\partial^2 f(x)}{\partial x_N^2} \end{bmatrix}_{\mathrm{x}=\mathrm{x}_0}$$

$$\mathrm{x}^* = \mathrm{x}_0 - H^{-1} \cdot \nabla f(\mathrm{x}_0)$$

- Hessian matrix is too big; hard to estimate
- Quasi-Newton’ s method: no need to compute Hessian matrix; quick update to approximate it.

Optimization Methods

- Convex optimization algorithms:
  - Linear Programming
  - Quadratic programming (nonlinear optimization)
  - Semi-definite Programming

- EM (Expectation-Maximization) algorithm.

- Growth-Transformation method.

Other Relevant Topics

- Statistical Hypothesis Testing
  - Likelihood ratio testing
- Linear Algebra:
  - Vector, Matrix;
  - Determinant and matrix inversion;
  - Eigen-value and eigen-vector
  - Derivatives of matrices;
  - etc.