MIE 1624 Introduction to Machine Learning and Data Analyzing

YIFAN ZHANG in 2018 FALL

Introduction to Analytics

- the scientific process from deriving insights from data in order to make decisions
- Descriptive analytics: what has happened?
- Predictive Analytics: what will happen?
- Perscriptive Analytics and AI: what should we do?

Python

- interpreted language not a complied (run code incrementally)
- List/Tuples/Strings are ordered collections

Dictionary:

- mapped(unordered) collections
- Syntax:

```
my_dictionary = {key_1:data_1, key_2:data_2, key_n:data_n}
```

```
clear() //del dict['Name']
copy()
fromkeys() //Create a new dictionary with keys from seq and values set to
value.
get(key, default=None)
has_key(key)
items() //Returns the data in the dictionary as a list of tuples (key,
value)
keys() //Returns the keys in the dictionary as a list
update(dict2) values()

for key, value in dict.items(): //iterate
```

Tuples

- similar to lists, but cannot be changed

Set

- efficient membership check

```
my_set = {1, 2, 3}
or:
x = [1,2,3]
my_set = set(x)
```

Python Lambda

- A lambda function is a small anonymous function.
- A lambda function can take any number of arguments, but can only have one expression.
- Syntax: lambda *arguments* : *expression*

```
x = lambda a : a + 10
print(x(5))
```

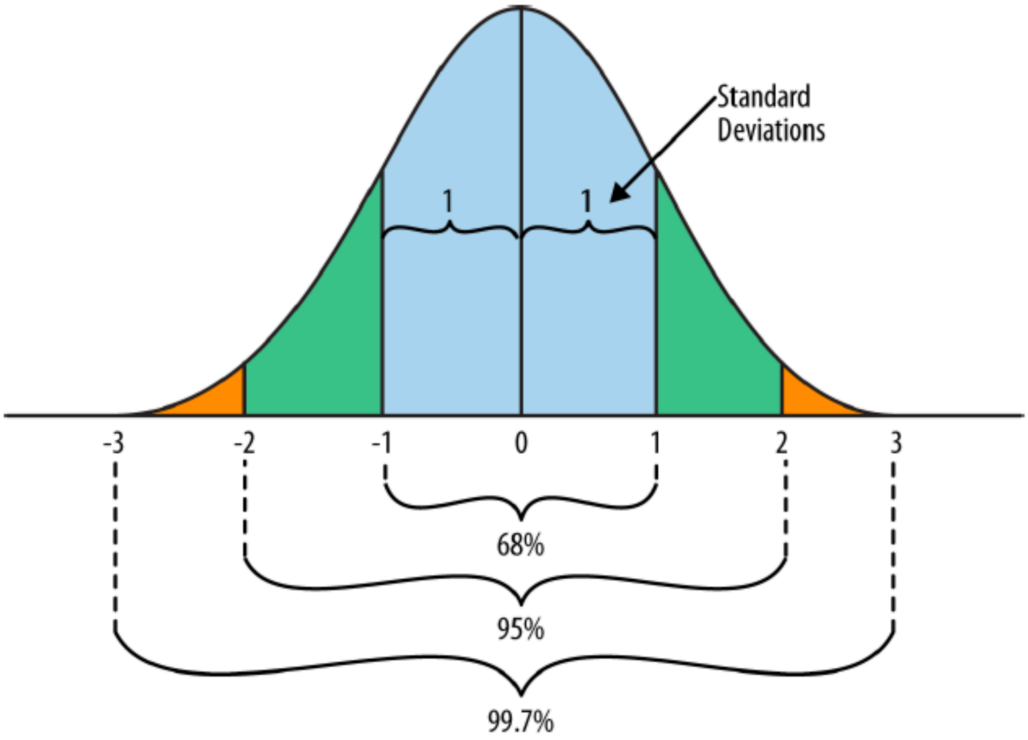
Pandas

- apply() applymap() map()
  - Apply() apply one function on all columns or rows
  - applymap() apply one function on each element with dataframe
  - map() apply one function on Series

```
get_first_letter = lambda x: x[0]
first_letters = names2010.name.map(get_first_letter)
```

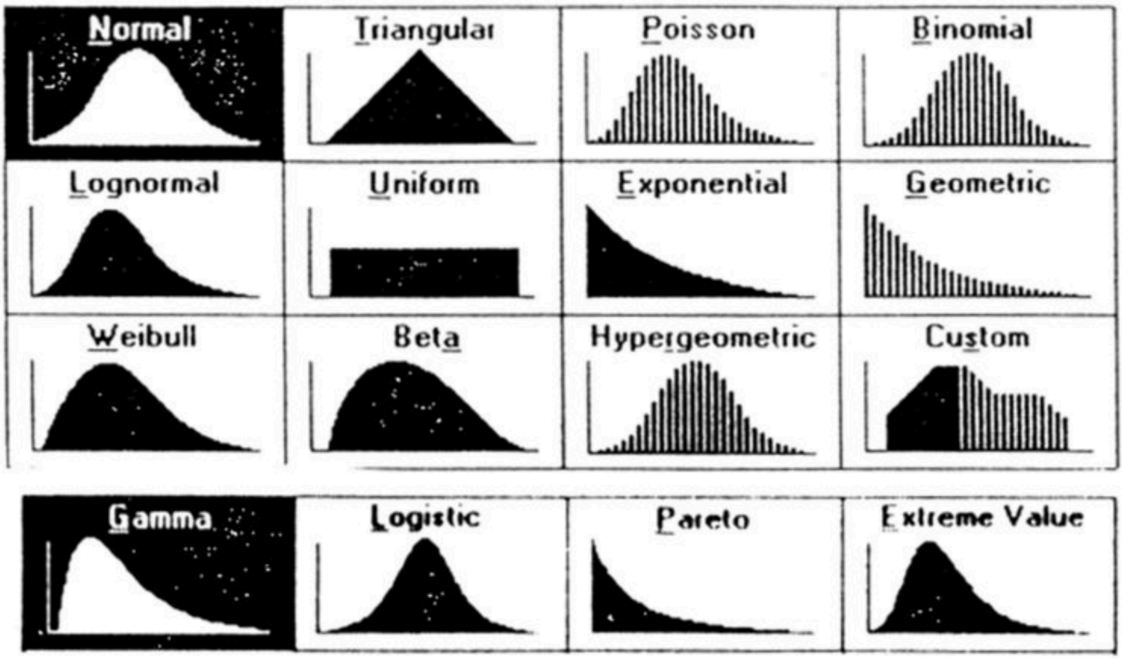
Basic Statistics(Need Review)

- What kind of data are we deal with: Quantitative or Categorical
  - Quantitative data:
    - mean
    - median
    - standard variace $s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}}$




```
- Percentile

- Confidence intervals(CI)

- Margin of Errors(MoE)

- T-test and statistically significant

  - if p-value < 0.05, statistically significant. We could conclude that the
difference is also present in unobserved population
  - If p-value > 0.05,  the difference observed could easily be simple due to
chance.
  - Make sure all your observations are truly independent (repeated
observations are cheating!
  - The larger the sample, the more likely the difference of a given size will
be significant
```

- Categorical data
- Distribution

- 

-

	Notation	$F_X(x)$	$f_X(x)$	$\mathrm{E}[X]$	$\mathrm{V}[X]$	$M_X(s)$
Uniform	$\mathrm{Unif}(a,b)$	$\begin{cases} 0 & x<a \\ \frac{x-a}{b-a} & a<x<b \\ 1 & x>b \end{cases}$	$\frac{I(a<x<b)}{b-a}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$	$\frac{e^{sb}-e^{sa}}{s(b-a)}$
Normal	$\mathcal{N}(\mu,\sigma^2)$	$\Phi(x)=\int_{-\infty}^{x}\phi(t)\,dt$	$\phi(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$	μ	σ^2	$\exp\left\{\mu s+\frac{\sigma^2 s^2}{2}\right\}$
Log-Normal	$\ln\mathcal{N}(\mu,\sigma^2)$	$\frac{1}{2}+\frac{1}{2}\,\mathrm{erf}\left[\frac{\ln x-\mu}{\sqrt{2\sigma^2}}\right]$	$\frac{1}{x\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(\ln x-\mu)^2}{2\sigma^2}\right\}$	$e^{\mu+\sigma^2/2}$	$(e^{\sigma^2}-1)e^{2\mu+\sigma^2}$	
Multivariate Normal	$\mathrm{MVN}(\mu,\Sigma)$		$(2\pi)^{-k/2}\lvert\Sigma\rvert^{-1/2}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$	μ	Σ	$\exp\left\{\mu^T s+\frac{1}{2}s^T\Sigma s\right\}$
Student's t	$\mathrm{Student}(\nu)$	$I_x\left(\frac{\nu}{2},\frac{\nu}{2}\right)$	$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-(\nu+1)/2}$	0	0	
Chi-square	χ^2_k	$\frac{1}{\Gamma(k/2)}\gamma\left(\frac{k}{2},\frac{x}{2}\right)$	$\frac{1}{2^{k/2}\Gamma(k/2)}x^{k/2}e^{-x/2}$	k	$2k$	$(1-2s)^{-k/2}\ s<1/2$
F	$\mathrm{F}(d_1,d_2)$	$I_{\frac{d_1 x}{d_1 x+d_2}}\left(\frac{d_1}{2},\frac{d_1}{2}\right)$	$\frac{\sqrt{\frac{(d_1 x)^{d_1}\,d_2^{d_2}}{(d_1 x+d_2)^{d_1+d_2}}}}{x\mathrm{B}\left(\frac{d_1}{2},\frac{d_1}{2}\right)}$	$\frac{d_2}{d_2-2}$	$\frac{2d_2^2(d_1+d_2-2)}{d_1(d_2-2)^2(d_2-4)}$	
Exponential	$\mathrm{Exp}(\beta)$	$1-e^{-x/\beta}$	$\frac{1}{\beta}e^{-x/\beta}$	β	β^2	$\frac{1}{1-\beta s}\ (s<1/\beta)$
Gamma	$\mathrm{Gamma}(\alpha,\beta)$	$\frac{\gamma(\alpha,x/\beta)}{\Gamma(\alpha)}$	$\frac{1}{\Gamma(\alpha)\beta^\alpha}x^{\alpha-1}e^{-x/\beta}$	$\alpha\beta$	$\alpha\beta^2$	$\left(\frac{1}{1-\beta s}\right)^\alpha\ (s<1/\beta)$
Inverse Gamma	$\mathrm{InvGamma}(\alpha,\beta)$	$\frac{\Gamma\left(\alpha,\frac{\beta}{x}\right)}{\Gamma(\alpha)}$	$\frac{\beta^\alpha}{\Gamma(\alpha)}x^{-\alpha-1}e^{-\beta/x}$	$\frac{\beta}{\alpha-1}\ \alpha>1$	$\frac{\beta^2}{(\alpha-1)^2(\alpha-2)^2}\ \alpha>2$	$\frac{2(-\beta s)^{\alpha/2}}{\Gamma(\alpha)}K_\alpha\left(\sqrt{-4\beta s}\right)$
Dirichlet	$\mathrm{Dir}(\alpha)$		$\frac{\Gamma\left(\sum_{i=1}^k\alpha_i\right)}{\prod_{i=1}^k\Gamma(\alpha_i)}\prod_{i=1}^k x_i^{\alpha_i-1}$	$\frac{\alpha_i}{\sum_{i=1}^k\alpha_i}$	$\frac{\mathrm{E}[X_i](1-\mathrm{E}[X_i])}{\sum_{i=1}^k\alpha_i+1}$	
Beta	$\mathrm{Beta}(\alpha,\beta)$	$I_x(\alpha,\beta)$	$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$	$1+\sum_{k=1}^{\infty}\left(\prod_{r=0}^{k-1}\frac{\alpha+r}{\alpha+\beta+r}\right)\frac{s^k}{k!}$
Weibull	$\mathrm{Weibull}(\lambda,k)$	$1-e^{-(x/\lambda)^k}$	$\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}e^{-(x/\lambda)^k}$	$\lambda\Gamma\left(1+\frac{1}{k}\right)$	$\lambda^2\Gamma\left(1+\frac{2}{k}\right)-\mu^2$	$\sum_{n=0}^{\infty}\frac{s^n\lambda^n}{n!}\Gamma\left(1+\frac{n}{k}\right)$
Pareto	$\mathrm{Pareto}(x_m,\alpha)$	$1-\left(\frac{x_m}{x}\right)^\alpha\ x\geq x_m$	$\alpha\frac{x_m^\alpha}{x^{\alpha+1}}\ x\geq x_m$	$\frac{\alpha x_m}{\alpha-1}\ \alpha>1$	$\frac{x_m^n}{(\alpha-1)^2(\alpha-2)}\ \alpha>2$	$\alpha(-x_m s)^\alpha\Gamma(-\alpha,-x_m s)\ s<0$

- Central Limit Theorem
  - Arithmetic means from a sufficiently large number of random samples from the entire population will be **Normally distributed** around the population mean (regardless of the distribution in the population)
  - if $E(x_i)=\mu$ and $var(x_i)=\sigma^2$ for all i then:

    $x_1+x_2+\ldots+x_n \sim N(n\bullet\mu, n\bullet\sigma^2)$

    $\bar{x}=\frac{x_1+\ldots+x_n}{n}\sim N(\mu,\sigma^2/n)$ When n increases, the distribution become more and more normal, and the spread of the distribution decreases.
  - 95% of the time, an individual sample mean should lie within 2 (actually 1.96) standard deviations of the mean

        - $prob[(\mu - 1.96s) \leq \mu \leq (\mu + 1.96s)] = 0.95$
        - $prob[(\mu - 1.96\frac{\sigma}{\sqrt{n}}) \leq \mu \leq (\mu + 1.96\frac{\sigma}{\sqrt{n}})] = 0.95$
        - **margin of error**: $1.96\frac{\sigma}{\sqrt{n}}$
    - Standard Deviation s of the sampling distribution of the mean of x is: $s^2 = \sigma^2/n$
    - Binominal distribution: $s = $
- Data collection: independent observation or dependent observation
    - Independent observations: one observation each object
    - dependent observations: repeated observation of same subject, relationship within groups, relationship over time or space
- PDF(probability density function) 概率密度函数
    - 如果X是连续型随机变量，定义概率密度函数为$f(X)$，用PDF在某一区间上的积分来刻画随机变量落在这个区间中的概率，即$P(a \leq X \leq b) = \int_a^b f_X(x)dx$
- PMF(probability mass function) 概率质量函数
    - 如果X离散型随机变量，定义概率质量函数为$f_X(x)$,PMF其实就是高中所学的离散型随机变量的分布律,即 $P(X = x) = f_X(x)$
- CDF(cumulative distribution function) 累计分布函数
    - 对于连续型随机变量， $F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_X(t)dt$
    - 对于离散型随机变量， 则为对应矩阵面积

Review of Linear Algebra

Gradients and Hessian Matrix

- The gradient of f at x, written as $\triangledown f(x) = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n})^T$
- The gradient vector $\triangledown f(x)$ gives the direction of steepest ascent of the function f at the point x
- Hessian matrix:

$$\triangledown^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_n^2} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Taylor series expansion

- First-order Taylpr series expansion: $f(x) = f(x_0) + \triangledown f(x_0)^T(x - x_0)$
- Second-order Taylor series expansion:
  $f(x) = f(x_0) + \triangledown f(x_0)^T(x - x_0) + \frac{1}{2}(x - x_0)^T \triangledown^2 f(x_0)^T(x - x_0)$

Newton Method

- $x^1 = x^0 - (\nabla^2 f(x^0))^{-1} \nabla f(x^0)$

Convex function

- Definition: a funciton is convex if for any $x^1, x^2 \in C \; and \; 0 \leq \lambda \leq 1$:
  $f(\lambda x^1 + (1-\lambda)x^2) \leq \lambda f(x^1) + (1-\lambda) f(x^2)$
- A square matrix A said to be positive definite(PD) if $x^T Ax > 0$ for all $x \neq 0$
- A square matrix A said to be positive semidefinite(PSD) if $x^T Ax > 0$ for all x
- Hessian $\nabla^2 f(x)$ is PD $\rightarrow$ strictly convex function
- Hessian $\nabla^2 f(x)$ is PSD $\rightarrow$ convex function
- Gradient $\nabla f(\bar{x}) = 0$ and Hessian $\nabla^2 f(x)$ is PD $\rightarrow \bar{x}$ is a minimum of the function f
- Gradient $\nabla f(\bar{x}) = 0$ and Hessian $\nabla^2 f(x)$ is PSD $\rightarrow \bar{x}$ is a strict minimum of the function f
- Check a matrix for PD or PSD
  - By computing principal minors
    - Leading principal minors $D_k, k = 1, 2, \ldots, n$ of a matrix $A = (a_{ij})_{n \times n}$ are defined as $D_k = \begin{vmatrix} a_{11} & \ldots & a_{1k} \\ \ldots & \ldots & \ldots \\ a_{k1} & \ldots & a_{kk} \end{vmatrix}$
    - A square matrix A is PD $\longleftrightarrow D_k > 0 \; for \; all \; k = 0, 1, 2, \ldots, n$
  - By computing its eigenvalues
    - $Ax - \lambda x = 0 \rightarrow (A - \lambda I)x = 0$, since s is non-zero, the determinant of $(A - \lambda I)$ should be vanish. Therefore, $det(A - \lambda I) = 0$
    - all eigenvalues are strictly positive, the Hessian is PD
- Properties of convex functions:
  - if f is a convex function, its sublevel $f(x) \leq \alpha$ is convex
  - $f \; convex, \; \alpha \geq 0 \rightarrow \alpha f \; convex$
  - $f_1, f_2 \;\; convex \rightarrow f_1 + f_2 \; convex$
  - $f_1, f_2 \;\; convex \rightarrow \max(f_1, f_2) \; convex$
  - $f \; convex \rightarrow f(Ax + b) \; convex$
- Composition rules:
  - $f(x) = h(g(x))$ Is convex if
    - g convex; h convex nondecreasing
    - g concave; h convex nonincreasing

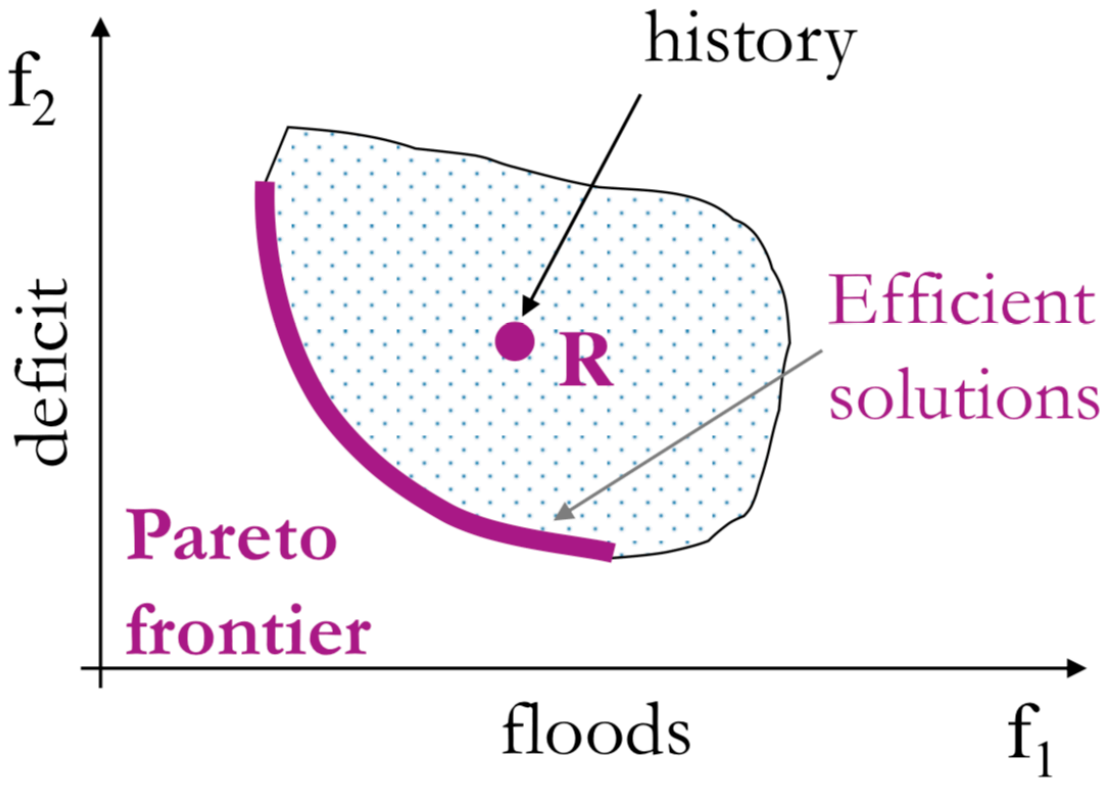
Modeling and Regressions

Modeling

- Model: simplified representation or abstraction of reality
- why we need a model?
  - building a model force detailed examination and thought about a problem, structures our thinking
  - searching for general insights
  - looking for specific numeric answers to a decision making problem
  - find the best way to do something
- seven ways to build a mode:
  - Define the problem
  - Observe the system, collect the data
  - Formulate models
  - Verify/validate model and use for prediction and exploration of system being modeled
  - Use model to help select among alternatives
  - Present results to decision makers
  - Implement solution and evaluate outcomes

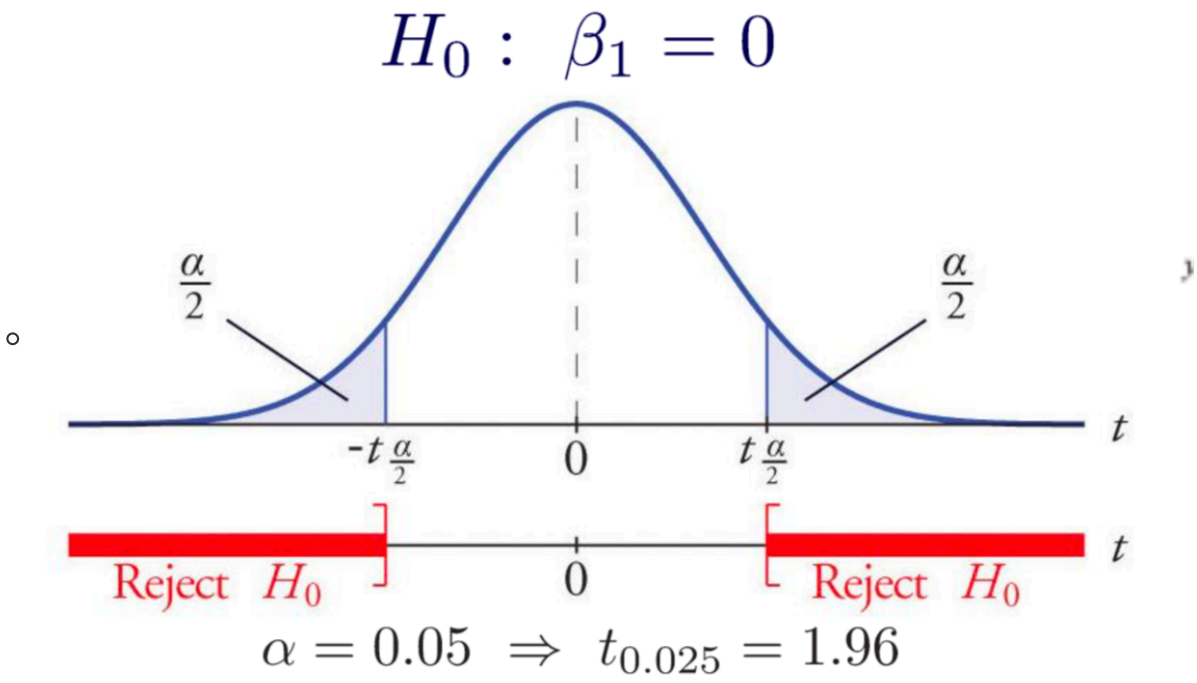
Decision Problem

- Elements
  - Objective f
  - Decision x
  - Constraints: feasible region Ω
- Transform the answers into equations
  - $min_x f(x)$
    $s.t. x \in \Omega$
- Multi-objective optimization: simultaneous optimizing two or more conflicting objectives subject to a certain constraints
  - $\min_{x \subset R} F(\mathbf{x}) = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)]$
    $s.t. x \in \Omega$
  - Define a indicator for each objective, to quantify the performance of different solutions
- Pareto efficiency: 帕雷托最优是指资源分配的一种理想状态。给定固有的一群人和可分配的资源，如果从一种分配状态到另一种状态的变化中，在没有使任何人境况变坏的前提下，使得至少一个人变得更好，这就是帕雷托改善。**帕雷托最优**的状态就是不可能再有更多的帕雷托改善的状态；换句话说，不可能在不使任何其他人受损的情况下再改善某些人的境况。
-



Linear Regression: statistical perspective

- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$
  - R-squared(adjusted R-squared), varies between 0 and 1, higher R-squared means a better fit
  - P-value: conclude a statistic significant finding when the p-value associated with a test is less than 0.05(above 95%)
  - F-test: taken together the exploratory variables in regression model are collectively different from 0
  - T-statistic(没懂)
- Hypothesis Testing:
  - Null hypothesis: no linear relationship exists between independent variable x and dependent variable y



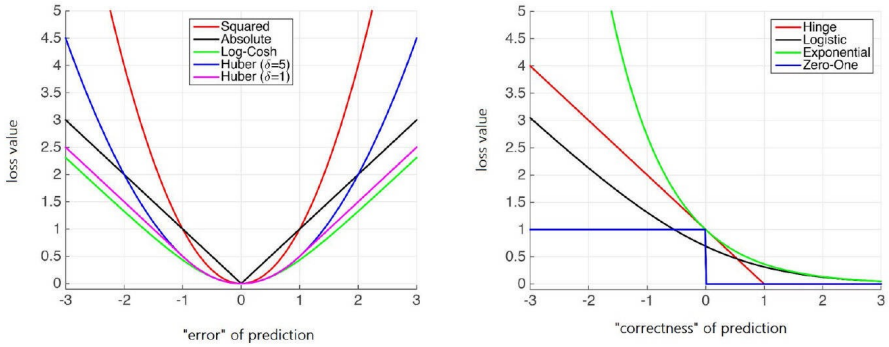
- Distribution of Errors: $\varepsilon \sim N(\mu, \sigma^2)$

Linear Regression: data science perspective

- Predict a value of a given continuous variable based on the values of other variables, assuming a linear or nonlinear model of dependency
- **Ussage**:
  - To predict, estimate or forecast the values: linear regression can be used to fit a predictive model to an observed data set of y and x values
  - To quantify the strength of the relationship between y and the x_i. To assess which x_j has a strong relationship or whether a particular x_k has a statistically significant relationship with a target variable.
- **Prediction**: models are numeric/ continuous/ Ordered valued
  - Split your data into training and test set
  - Construct a model using training set
  - Evaluate your model using test set
  - Use model to predict unknown value
- Prediction algorithms:
  - Regression
    - Simple regression $y = \beta_1 x + \beta_0$
    - Multiple regression $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n$
    - Linear regression
    - Non-linear regression $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 + \beta_3 x_3^3 + \ldots + \beta_n x_n^n$
  - k-nearest neighbor methods – Neural Networks
  - Support Vector Regression

- Least Squares
  - Linear regression fits a model with coefficient $\beta = (\beta_1, \beta_2, \beta_3, \ldots, \beta_n)^T$ to minimize the residual sum of squares between the observed response(y) in dataset, and response predict by the linear approximation ($\hat{y}$) $\min_\beta ||X\beta - y||_2^2$
- Accuracy
  - Loss function:
    - Absolute Error: $|y^{(j)} - \hat{y}^{(j)}|$
    - Squared Error: $(y^{(j)} - \hat{y}^{(j)})^2$

Loss Function	Used For	Comments	Advantages	Disadvantages
Squared Loss (OLS)	**Regression**	Estimates **mean** label; most popular regression loss function.	Differentiable everywhere.	Sensitive to outliers / noise.
Absolute Loss	**Regression**	Estimates the **median** label.	Less sensitive to noise than OLS.	Not differentiable at 0.
Huber Loss (Smooth Absolute Loss)	**Regression**		"Best of both worlds" of squared and absolute loss. Takes on behavior of squared loss when loss is small, and absolute loss when loss is large.	Once differentiable.
Log-Cosh Loss	**Regression**		Similar to Huber Loss, but twice differentiable everywhere.	
Hinge Loss	**Classification**	Standard SVM (p=1) (Differentiable) Squared Hingeless SVM (p=2)		Standard SVM is only differentiable everywhere at p=2.
Log-Loss	**Classification**	As $z \to -\infty$, log-loss and hinge loss become increasingly parallel.	Outputs are well-tuned; popular.	
Exponential Loss	**Classification**	Exponential loss and hinge loss are both upper-bounds of zero-one loss.		Loss increases exponentially - extremely aggressive.
Zero-One Loss	**Classification**	Loss is zero when the prediction is correct, and one when the prediction is incorrect.		Non-continuous, which means it's not really practical to optimize.



  - Test Error: the average lost over test set
    - Mean Absolute Error: $\frac{\sum_{j=1}^{m} |y^{(j)} - \hat{y}^{(j)}|}{m}$
    - Mean Squared Error: $\frac{\sum_{j=1}^{m} (y^{(j)} - \hat{y}^{(j)})^2}{m}$
    - Relative Absolute Error: $\frac{\sum_{j=1}^{m} |y^{(j)} - \hat{y}^{(j)}|}{\sum_{j=1}^{m} |y^{(j)} - \bar{y}|}$
    - Relatve Square Error: $\frac{\sum_{j=1}^{m} (y^{(j)} - \hat{y}^{(j)})^2}{\sum_{j=1}^{m} (y^{(j)} - \bar{y})^2}$
    - R^2 = 1 - Relative Square Error
      - measures how well the regression line approximates the real data points, it also portrays **percent of variance in the data explained by regression model**
        - If the value is close to 1, the model fits perfectly and explains all variance
        - If the value is close to 0, then the model does not fit the data and doesn't explain any

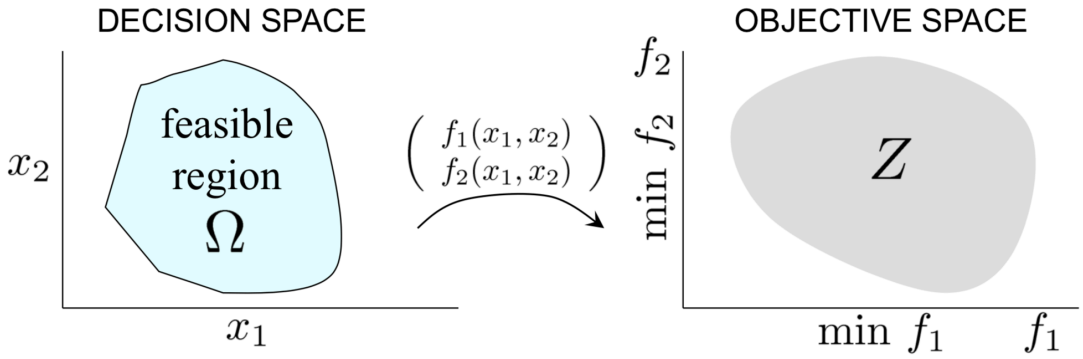
      variance

- Variable preparation and selection
  - Preparation
    - Interval variables can be binned or bucketed in order to capture nonlinear relationship
    - Categorical variables must be converted into binary vectors. Data sample must be large enough to accommodate all degrees of freedom

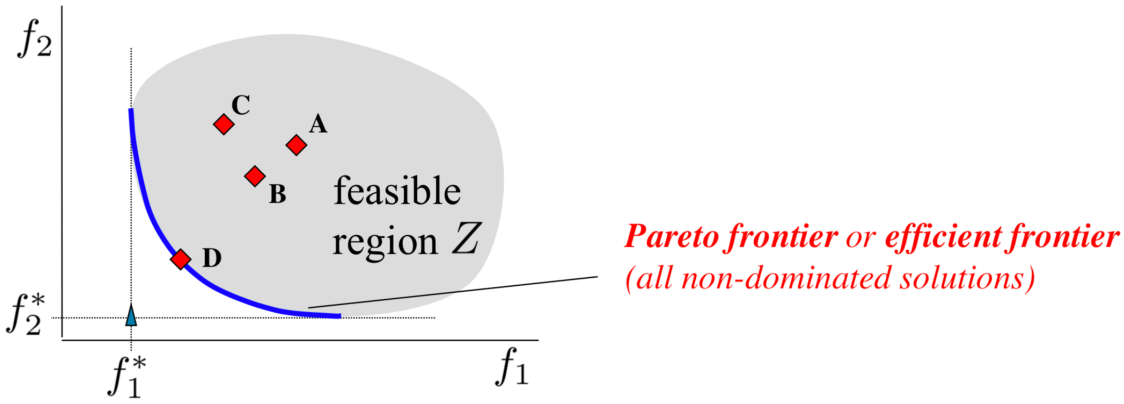
      > Norminal Data 定类变量：变量的不同取值仅仅代表了不同类的事物，这样的变量叫定类变量。问卷的人口特征中最常使用的问题，而调查被访对象的“性别”，就是 定类变量。对于定类变量，加减乘除等运算是没有实际意义的。
      >
      > Ordinal Data定序变量：变量的值不仅能够代表事物的分类，还能代表事物按某种特性的排序，这样的变量叫定序变量。问卷的人口特征中最常使用的问题“教育程度“，以及态度量表题目等都是定序变量，定序变量的值之间可以比较大小，或者有强弱顺序，但两个值的差一般没有什么实际意义。
      >
      > Interval Data 定距变量：变量的值之间可以比较大小，两个值的差有实际意义，这样的变量叫定距变量。有时问卷在调查被访者的“年龄”和“每月平均收入”，都是定距变量。
      >
      > Ratio Data 定比变量，有绝对0点，如质量，高度。定比变量与定距变量在市场调查中一般不加以区分，它们的差别在于，定距变量取值为“0”时，不表示“没有”，仅仅是取值为0。定比变量取值为“0”时，则表示“没有”。
  - Variable Selection (LASSO Algorithm)
    - $\min_\beta ||X\beta - y||_2^2 + \lambda||\beta_1||$
    - $\min_\beta ||X\beta - y||_2^2$ Subject to $||\beta_1|| \leq \varepsilon$
- Bias-Variance trade off
  - Prediction Error for new data $|\tilde{X}\beta - y||_2^2$ should be very small:
    - $PE(\tilde{X}) = \sigma_i^2(irreducible\ error) + Bias(\tilde{X}\beta) + Vars(\tilde{X}\beta)$
    - $PE(\tilde{X}) = \sigma_i^2 + (E[\hat{y}] - y)^2 + E[\hat{y} - E[\hat{y}]]^2$
    - Bias-Variance Trade-off:
      - Bias: How much predicted value differ from true value
      - Variance: How predictions made on the same value vary from different realizations of the model
      - As model becomes more complex (more terms included), model structure can be picked up. While coefficient estimates suffer from high variance as more terms are included in the model.

Muti-objective optimization

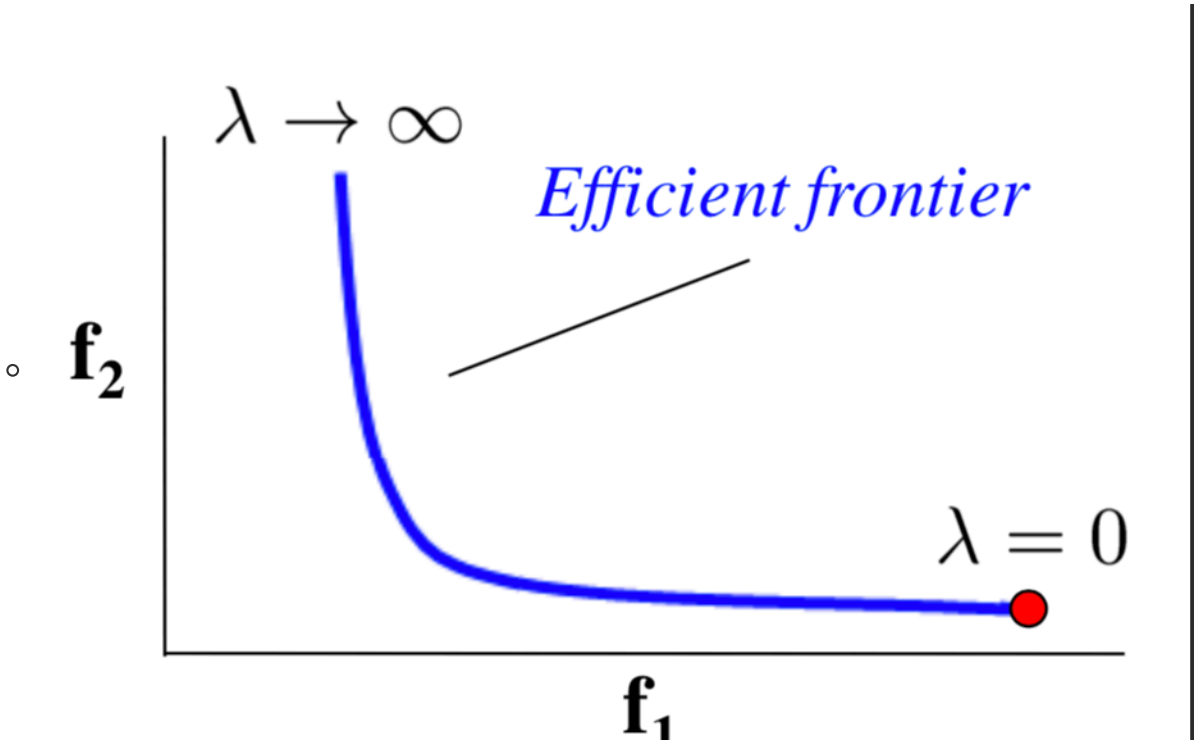
- $\min_{s.t.x\in\Omega}(f_1(x), f_2(x), \ldots, f_k(x))$
- Mapping feasible region into objective space
-



- Example: $\min f_1 = bias$, $\min f_2 = variance$
- Pareto efficiency: solutions with characteristics like D, are called tradeoff, Pareto optimal or non-dominated
-



- Multi-objective optimization goal: find solution(s) **on the efficient frontier** according to the decision maker preferences
- Computing efficient frontier
  - Ideal goal: compute exact frontier
  - Typical goal: approximate the frontier
- Solving multi-objective optimizaiton problem
  - Convert multi-objective optimization problem to a series of single-objective optimization problems
- Weighted Method
  - Assign weights to each objective and Optimize the weighted sum of the objectives
  - $\min_{s.t.x\in\Omega} w_1 \bullet f_1(x) + w_2 \bullet f_1(x)$, $\lambda = \frac{w_1}{w_2}$

- 

- $\varepsilon-$Constraint(Hierarchical) Method
  - Optimize one objective and Convert other objectives into constraints

- First step

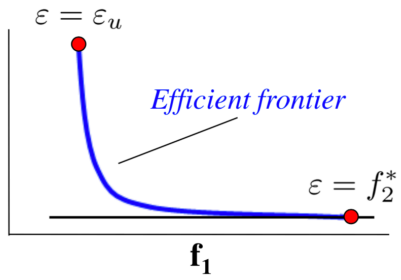
$$\begin{aligned} \min \quad & f_2(\boldsymbol{x}) \\ \text{s.t} \quad & \boldsymbol{x} \in \Omega \end{aligned}$$

$$\Downarrow$$

$$f_2^*$$

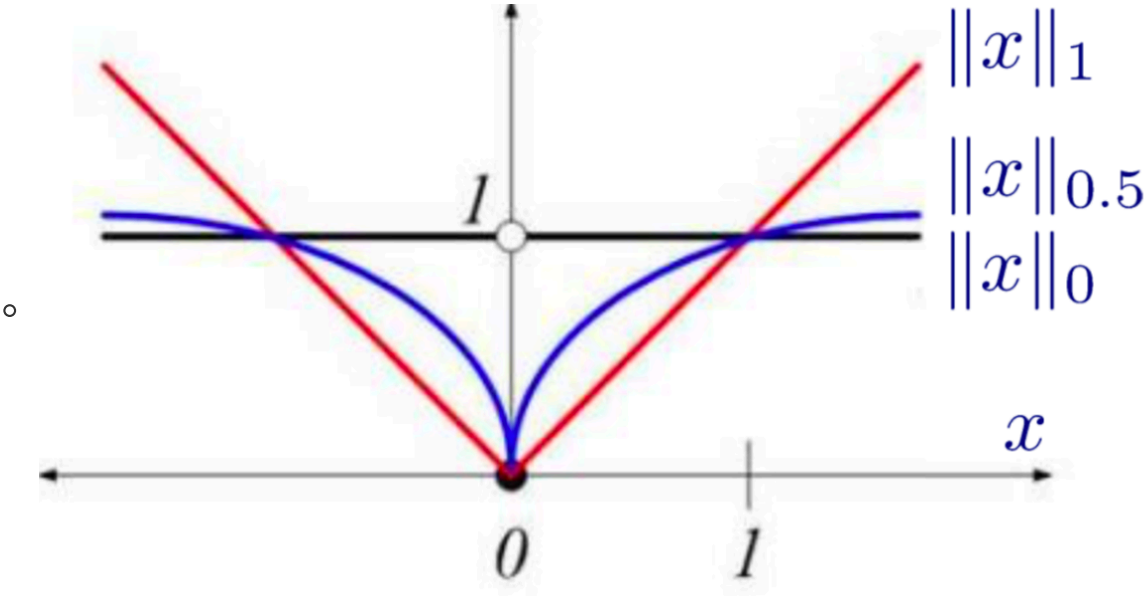
Second step

$$\begin{aligned} \min \quad & f_1(\boldsymbol{x}) \\ \text{s.t} \quad & \boldsymbol{x} \in \Omega \\ & f_2(\boldsymbol{x}) \leq \underbrace{(1+\epsilon) f_2^*}_{\varepsilon} \end{aligned}$$

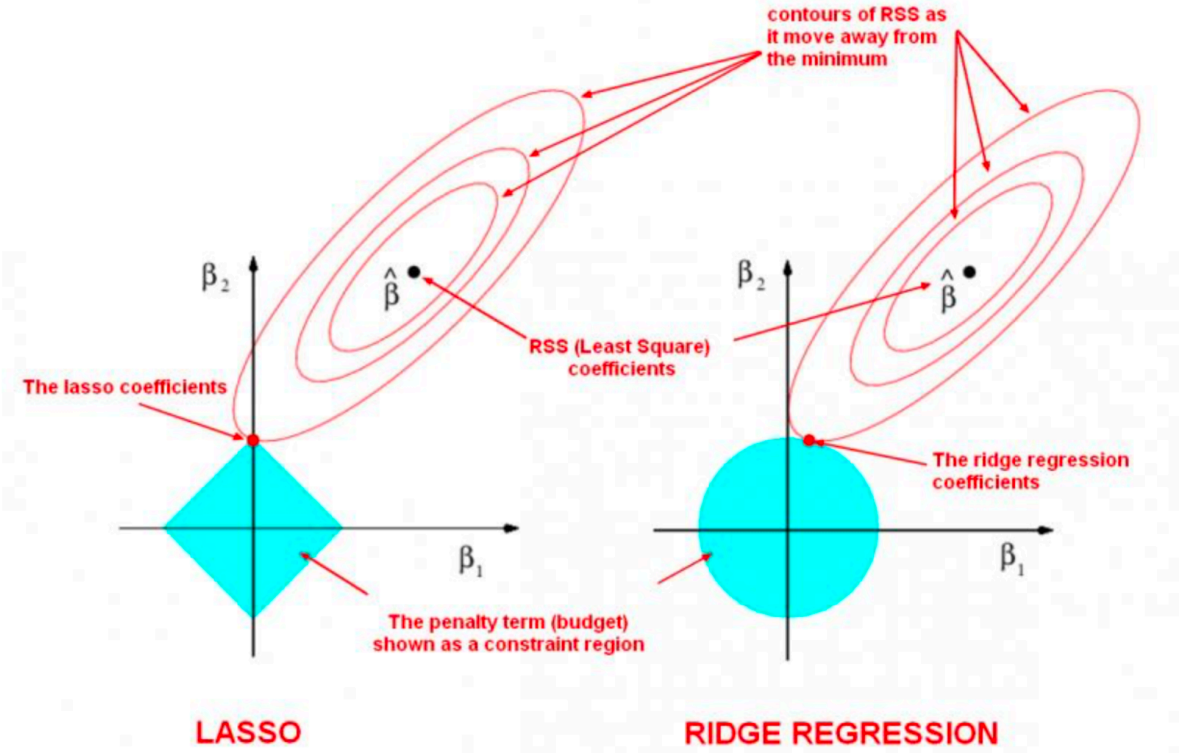


Regularized Regression

- Norm function: $||\mathbf{x}||_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p} = \mathbf{x}^T\mathbf{x}(whenp = 2)$

- 

- l_1 regularized regression and variable selection(LASSO algorithm)
  - $\min_\beta ||X\beta - y||_2^2 + \lambda||\beta_1||$
  - $\min_\beta ||X\beta - y||_2^2$, subject to $||\beta||_1 \leq \varepsilon$
  - $min_\beta (X\beta - y)^T (X\beta - y)$, subject to $\sum_{i=1}^n |\beta_i| \leq \varepsilon$
  - LASSO not have a closed-form solution:
    - $\beta^T (X^T X)\beta - 2(X^T y)^T \beta + y^T y$, subject to
    - $\sum_{i=1}^n |\beta_i| \leq \varepsilon$, where $\bar{\beta}_i + \underline{\beta_i} = |\beta_i|$, $\beta_i = \bar{\beta}_i - \underline{\beta_i}$, $\bar{\beta}_i \geq 0, \underline{\beta_i} \geq 0$
- l_2 regularized regression or Tikhonov regularization(Ridge regression)
  - $\min_\beta ||X\beta - y||_2^2 + \lambda||\beta_1||_2^2$
  - $\min_\beta ||X\beta - y||_2^2$, subject to $||\beta||_2^2 \leq \varepsilon$
  - $min_\beta (X\beta - y)^T (X\beta - y)$, subject to $\sum_{i=1}^n |\beta_i|^2 \leq \varepsilon$
  - Ridge regression has closed form solution: $\beta = (X^T X + \lambda I)^{-1} X^T y$
- When using LASSO, the solution is more likely to be at the corner, which contains 0 in parameters and cause sparse matrix(using for feature selection)
- When using RIDGE, the solution is more likely to be the average one.
-



- Tuning hyperparameters
  - l_1 regularized linear regression: $\min_\beta \frac{1}{m}||X\beta - y||_2^2 + \alpha||\beta_1||_1$
  - l1 regularized logistic regression: $min_\theta||\theta||_1 + C \bullet J(\theta)$
- Evaluation Method:
  - Handsout
    - Given data is randomly partitioned into two independent sets
      - Training set (e.g., 2/3) for model construction
      - Test set (e.g., 1/3) for accuracy estimation
    - Random sampling: a variation of holdout
      - Repeat holdout k times, accuracy = avg. of the accuracies obtained
  - Cross-Validaiton
    - Randomly partition the data into k mutually exclusive subsets, each approximately equal size
    - At i-th iteration, use Di as test set and others as training set.
    - Leave-one-out: k folds where k = # of tuples, for small sized data
    - Stratified cross-validation: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data 每个fold都是按照类别的比例抽出来的, 比如这个分类任务一共有三个类别A、B、C，它们的比例是1:2:10。那么每个fold中的A、B、C的比例也必须是1:2:10.

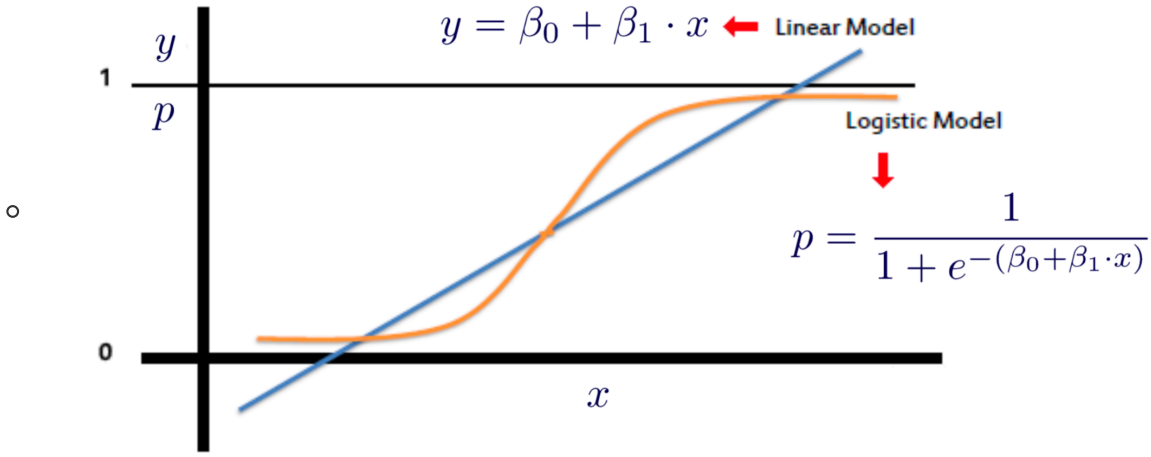
Linear Regression -- Machine Learning Perspective

- input(features)

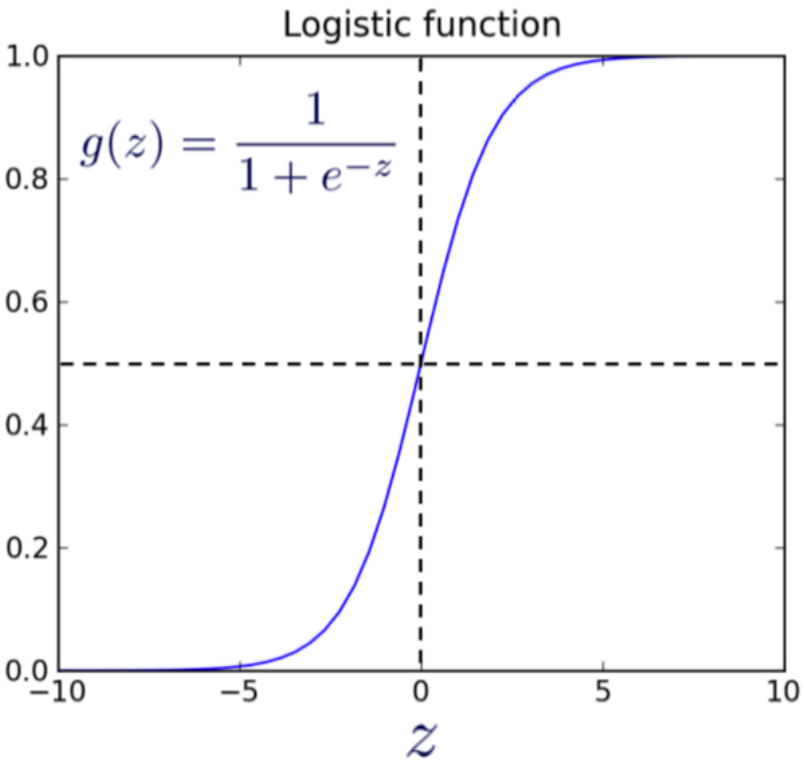
    - n -- number of features
    - $x^{(j)}$ -- input of j-th training example
    - $x_i^{(j)}$ -- value of feature i in j-th training example
    - $x^T = (x_0, x_1, \ldots, x_n)^T$
- Hypothesis
    - $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n, where\ x_0 = 1$
- Parameters
    - $\theta^T = (\theta_0, \theta_1, \ldots, \theta_n)^T$
- Cost function
    - $J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(j)}) - y^{(j)})^2$
- Optimization
    - $\min_\theta J(\theta)$
- Solving algorithms
    - Non-linear optimization methods, e.g., iterative algorithms such as gradient descent algorithms, Newton and Quasi-Newton algorithms, etc.
    - Linear algebra (normal equations), i.e., solving $\triangledown J(\theta)$, m samples, n features
        - $\triangledown J(\theta) = (\frac{\partial J(\theta)}{\partial \theta_0}, \frac{\partial J(\theta)}{\partial \theta_1} \ldots, \frac{\partial J(\theta)}{\partial \theta_n})^T$
        - $\frac{\partial J(\theta)}{\partial \theta_i} = \frac{1}{m} \sum_{j=1}^{m} (h_\theta(x^{(j)}) - y^{(j)}) \bullet x_i^{(j)} = 0$
        - $\sum_{j=1}^{m} (\theta^T x^{(j)} - y^{(j)}) \bullet x_i^{(j)} = 0$
        - $\sum_{j=1}^{m} \theta^T x^{(j)} \bullet x_i^{(j)} = \sum_{i=1}^{m} y^{(j)} \bullet x_i^{(j)}$
        - $\theta = (X^T X)^{-1} X^T y$ or. $(X^T X) \bullet \theta = X^T y$

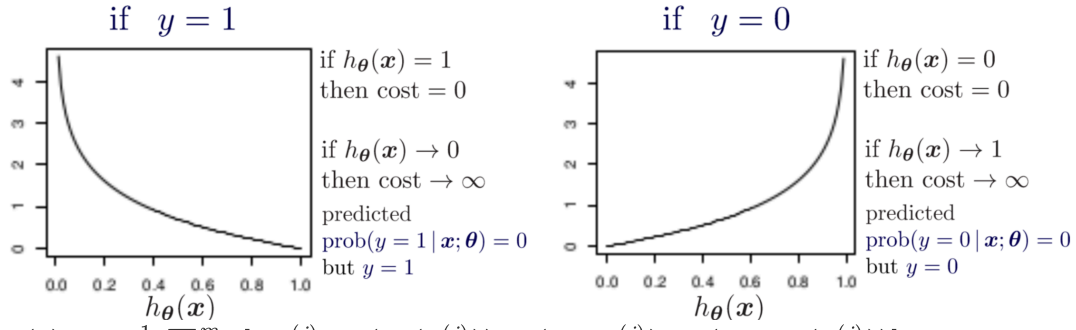
Logistic Regression -- Machine Learning Perspective

- Quantile Regression
    - Ordinary least squares regression approximates the conditional mean of the response variable, while quantile regression is estimating either the conditional median or other quantiles of the response variable
    - This is very helpful in case of skewed data (e.g., income distribution in the US) or to deal with data without suppressing outliers
- Logistic Regression
    - predict categorical target variable, most often a variable with a binary outcome
    - Logit and Probit regressions can also be used to predict binary outcome. While the underlying distributions are different, all three models will produce rather similar outcomes
    - Difference: **Linear regression** predicts real values, while **Logistic regression** predicts values in the range of 0 to 1

- 

- Classification
  - Features are numerical attributes, predict discrete targets
  - Classifer
    - $h(x|\theta) = h_\theta(x)$
  - Predictions
    - $\hat{y} = h_\theta(x)$
  - Threshhold:
    - If $h_\theta(x) \geq 0.5$, predict y = 1
    - If $h_\theta(x) \leq 0.5$, predict y = 0
  - We want $0 \leq h_\theta \leq 1$:
    - $h_\theta(x) = g(\theta^T x) = \frac{1}{1-e^{-\theta^T x}}$, by using sigmoid function/logistic function (differentiable)

- 

    - given x, estimate the possibility that $y = 1$, $h_\theta(x) = g(\theta^T x) = prob(y = 1; x, \theta)$
  - Cost function:
    - $J(\theta) = \frac{1}{m} \sum_{j=1}^{m} cost(h_\theta(x^{(j)}) - y^{(j)})$
    - $cost(h_\theta(x^{(j)}) - y^{(j)}) = \begin{cases} -\log(h_\theta(x)), y = 1 \\ -\log(1 - h_\theta(x)), y = 0 \end{cases}$
    - Compact form: $cost(h_\theta(x), y) = -y \log(h_\theta(x)) + (1 - y) \log(1 - h_\theta(x))$
    - 

    - $J(\theta) = -\frac{1}{m} \sum_{j=0}^{m} [-y^{(j)} \log(h_\theta(x^{(j)})) + (1 - y^{(j)}) \log(1 - h_\theta(x^{(j)}))]$

Gradient boosting for regression

- Recursive gradient descent
  - At each iteration gradient descent algorithm is used to fit a model to the residuals of the previous model
  - Bagging – fit a weak learner to a bootstrapped sample of dataset
  - Boosting – sequentially fit a weak learner to the residuals of the previous model
- Models:
  - Regression: $\hat{y_i} = beta_1 x_i + \beta_0$

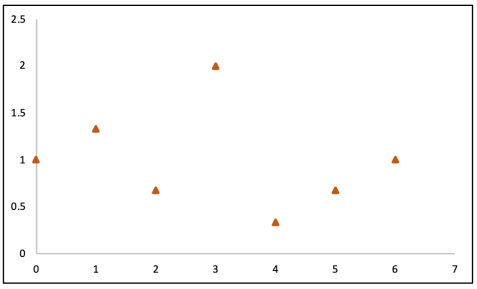
- cost function: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- Gradient descent: $\beta_j = \beta_{j-1} - \alpha \nabla MSE$
- $\nabla MSE = \left[\frac{-2}{n}\sum_{i=1}^{n} x_i(y_i - (\beta_1 x_i + \beta_0)) \quad \frac{-2}{n}\sum_{i=1}^{n}(y_i - (\beta_1 x_i + \beta_0))\right]$

- Algorithm:
  - Start with an arbitrarily chosen line as the original model
  - calculate the residual for current prediction

MSE: 1.25

x	y	Prediction	Residual
0	1	0	1
1	3	1.67	1.33
2	4	3.33	0.67
3	7	5	2
4	7	6.67	0.33
5	9	8.33	0.67
6	11	10	1

Residuals



  - using residuals of previous model as y and choose initial model for residual(prediction)

Initial model for residuals

$\beta_1^0 = 1$

$\beta_2^0 = 1$

$y - \tilde{y} = x + 1$

x	y (residuals of previous Model)	Prediction	Residual
0	1	1	0
1	1.33	2	-0.67
2	0.67	3	-2.33
3	2	4	-2
4	0.33	5	-4.67
5	0.67	6	-5.33
6	1	7	-6

MSE: 13.7

  - Iteration and find best parameters for the residual model
  - Add up original model and model for residuals

Data Mining and Machine Learning

- Machine Learning gives computer the ability to learn without being explictly programmed

Supervised Learning

- decision trees, ensembles (bagging, boosting, random forests), k-NN, linear regression, Naive Bayes, neural networks, logistic regression, SVM

Regression

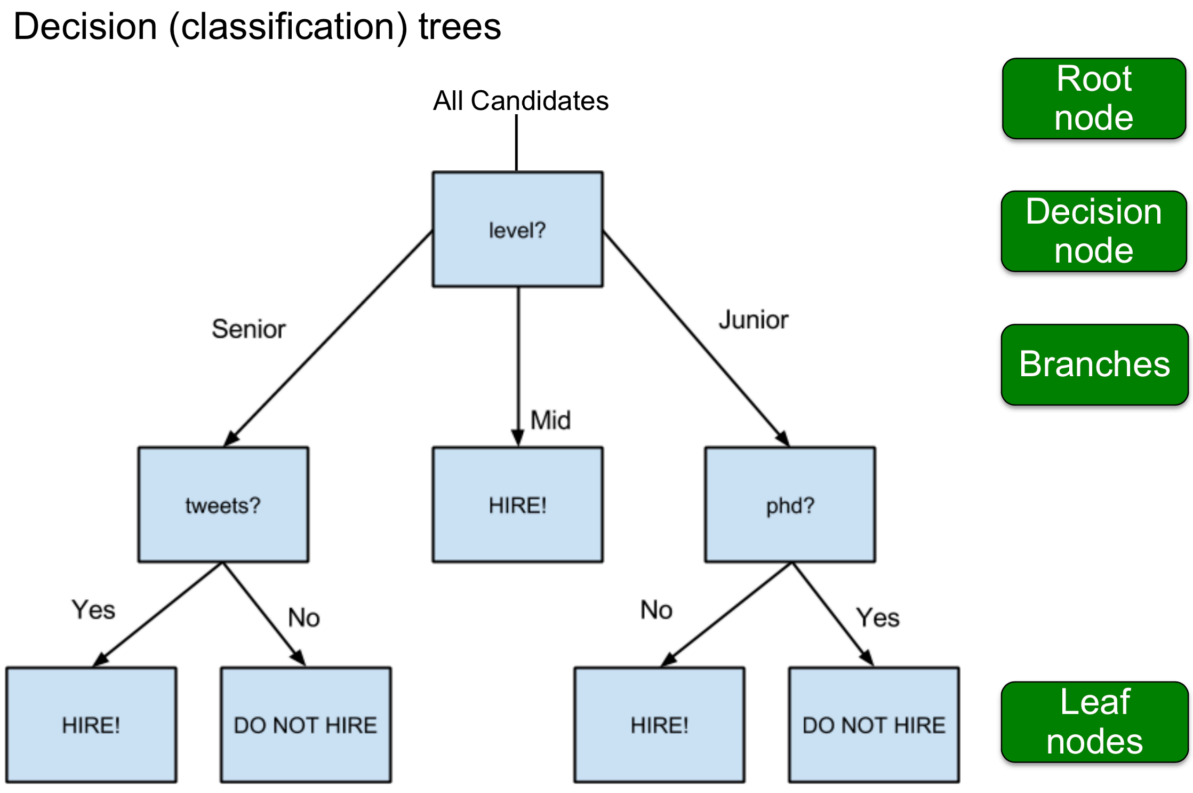
Classification

- Classification is a supervised learning technique, which maps data into predefined classes or groups
- Work with both interval and categorical features

Decision Tree

- A tree can be "learned" by splitting the source set into subsets based on an attribute value test
- Tree partitions samples into mutually exclusive groups by selecting the best splitting attribute, one group for each terminal node
- The process is repeated recursively for each derived subset, until the stopping criteria is reached
-

Pros	Cons
no need to normalize data	Easy to overfit or underfit the model
Easy to interpret	Cannot model interactions between features
Can handle numeric or categorical features, missing data	Large trees can be difficult to interpret
Uses only the most important features	The tree is no stable
Can be used on very large or small data	

- 

- Using recursive partitioning to classify data, chooses the most predictive feature to split the data("**Predictiveness" is based on decrease in entropy (gain in information) or "impurity**")

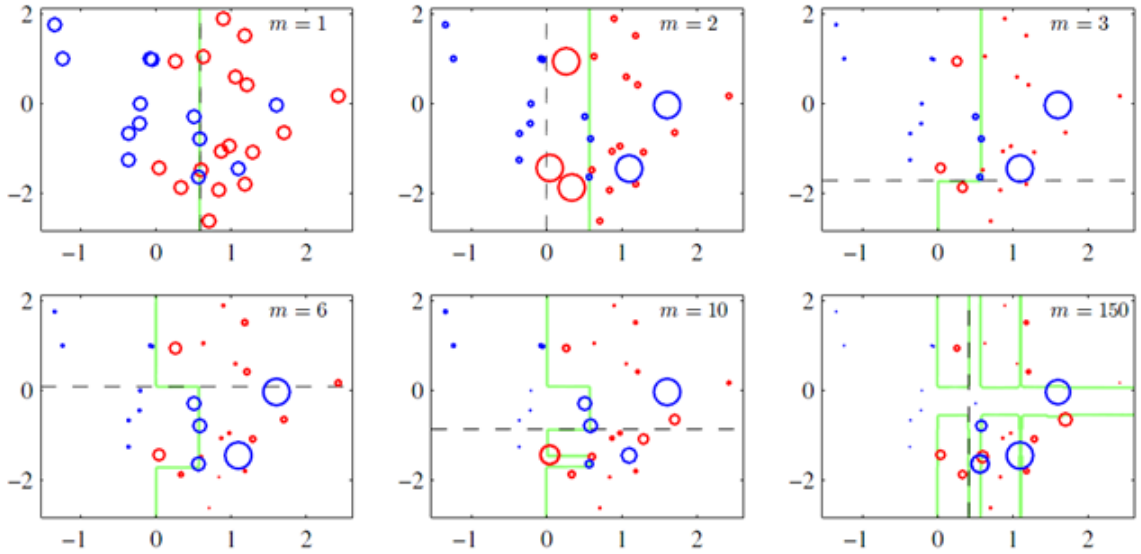
- **Information gain =entropy(parent) - average entropy(children)**
- Entropy is a common measure of target class impurity (i is each of the target classes, p_i is proportion of the number of elements in class 0 or 1):

  $Entropy = \sum_i -p_i \log_2(p_i)$
- Gini Index is another measure of impurity

  $Gini = 1 - \sum_i p_i^2$

Ensemble Learning

- 在机器学习的有监督学习算法中，我们的目标是学习出一个稳定的且在各个方面表现都较好的模型，但实际情况往往不这么理想，有时我们只能得到多个有偏好的模型（弱监督模型，在某些方面表现的比较好）。集成学习就是组合这里的多个弱监督模型以期得到一个更好更全面的强监督模型，集成学习潜在的思想是即便某一个弱分类器得到了错误的预测，其他的弱分类器也可以将错误纠正回来。
- 集成学习在各个规模的数据集上都有很好的策略。
  - **数据集大：划分成多个小数据集，学习多个模型进行组合**
  - **数据集小：利用Bootstrap方法进行抽样，得到多个数据集，分别训练多个模型再进行组合**
- Bagging = bootstrap aggregating
  - boostrap: 有放回的抽样方法
    - 采取重抽样的方法从原始样本中抽取一定数量的样本
    - 根据抽取的样本计算统计量T
    - 重复上述N次，一般N≥1000
    - 根据N个统计量T，计算出统计量的置信区间
  - bagging: 建立N个模型
    - 分类问题采用N个模型预测投票的方式
    - 回归问题采用N个模型预测平均的方式
  - Random Forest:
    - develop many unrelated decision trees(low bias, high variance), by sampling on both columns and rows
      - for rows, 采用有放回的方式，若有N个数据，则采样出N个数据（可能有重复），这样在训练的时候每一棵树都不是全部的样本，相对而言不容易出现overfitting
      - for columns: 从M个feature中选择出m个（m<<M）
      - 随机森林中的每一棵树的都对输入进行预测，最后进行投票，哪个类别多，输入样本就属于哪个类别
    - The trees are made uncorrelated to maximize the decrease in variance, but the algorithm cannot reduce bias (which is slightly higher than the bias of an individual tree in the forest)
- Boosting
  - 采用分层学习，通过m个步骤最终得到F
  -



- 上图（图片来自prml p660）就是一个Boosting的过程，绿色的线表示目前取得的模型（模型是由前m次得到的模型合并得到的），虚线表示当前这次模型。每次分类的时候，会更关注分错的数据，上图中，红色和蓝色的点就是数据，点越大表示权重越高，看看右下角的图片，当m=150的时候，获取的模型已经几乎能够将红色和蓝色的点区分开了。
- Gradient Boosting (based on weak learners (high bias, low variance))(**算法没懂**)
  - 每一次建立模型是在之前建立模型损失函数的梯度下降方向
  - **Algorithm 1: Gradient_Boost**

    $F_0(\mathbf{x}) = \arg\min_\rho \sum_{i=1}^{N} \Psi(y_i, \rho)$

    For $m = 1$ to M do:

    $\tilde{y}_i = -\left[\frac{\partial \Psi(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \; i = 1, N$

    $\mathbf{a}_m = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{N} [\tilde{y}_i - \beta h(\mathbf{x}_i; \mathbf{a})]^2$

    $\rho_m = \arg\min_\rho \sum_{i=1}^{N} \Psi(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m))$

    $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m)$

    endFor

    end Algorithm

- Bias-Variance Trade-off
  - the problem of simultaneously minimizing two sources of error that prevent supervised learning algorithms from generalizing beyond their training set
  - Bias is error from erroneous assumptions in the learning algorithm, high bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting)
  - Variance is error from sensitivity to small fluctuations in the training set, high variance can cause overfitting, i.e., modeling the random noise in the training data, rather than the intended outputs

Unsupervised Learning

- k-means, c-means, hierarchical clustering, DBSCAN
- Unlabeled data and no “target” variable

K-means Clustering

- Algorithm:
  - Randomly assign each of sample to K user specified clusters
  - compute average of each cluster(centroid)
  - compute the distance between each sample x_i and each centroid
  - Assign x_i to nearest centroid and recaculate the centroid of affected clusters
  - iterate until no more reasignments are made
- Intra-cluster distance are minimized, inter-cluster distance are maximized

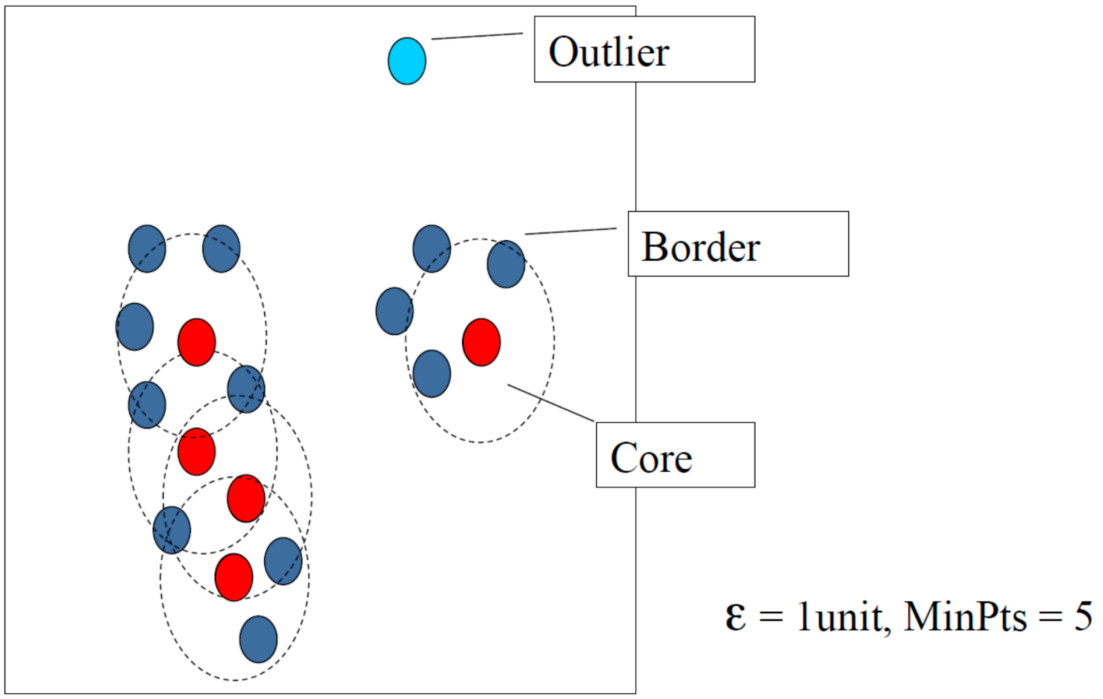
Fuzzy C-means clustering (FCM)

- allow one data belongs to two or more clusters
- Always converges, clustering noisy samples

Hierarchical clustering

- Hierarchica clustering are organized as trees where each node is the cluster consisting of the cluster of its daughter nodes
- Combine two nodes with smallest distance as a higher hierarchical at each round

DBSCAN Density Based Clustering

- locates the region of high density(number of point within a specified radius) that are seperate from one another by region of low density
- 

Association Rule

- Market basked analysis
- 记录出现次数，大于10记录
- 如果组合大于10次，也记录下来
- 得到最大的集合$\rightarrow$认为经常出现，大于既定confidence

Dimensionality Deduction

- PCA, LDA, factor analysis, t-SNE

Reinforcement Learning

- Dynamic programming

Association Rule

- Market basked analysis
- Detects associations (affinities) between variables (items or events)

Netural Nets

- deep learning, multilayer perceptron, recurrent neural network (RNN), convolutional neural network (CNN)
- Model is an assembly of inter-connected neurons (nodes) and weighted links
- Each neuron applies a nonlinear function to its inputs to produce an output
- Output node sums up each of its input value according to the weights of its links

Overview of Optimizaition

- General form: $\min_x c^T x + \frac{1}{2} x^T Q x$

$$s.t.\ l \leq Ax \leq u$$
$$l_b \leq x \leq u_b$$

- Solving linear optimization problems
  - Minimizing convex quadratic (QP) objective function over a polyhedron (linear constraints)
  - Interior plot methods(barrier algorithm in CPLEX)
    - it reaches a best solution by traversing the interior of the feasible region
    - slower: small num variables interior point method, large num variables simplex method
  - Simplex-type methods
    - it examines the adjacent vertices in sequence within feasible set to ensure that at each vertex the objective function is increasing or is non-affected

- graphical view:

$$\begin{aligned}\max_{x\in\mathbb{R}^3}\quad & 3x_1+2x_2+2x_3\\ \text{s.t.}\quad & x_1++x_3\le 8\\ & x_1+x_2\le 7\\ & x_1+2x_2\le 12\\ & x_1,\,x_2,\,x_3\ge 0\end{aligned}$$

$$\begin{aligned}\max_{x\in\mathbb{R}^3}\quad & 3x_1+2x_2+2x_3\\ \text{s.t.}\quad & x_1++x_3+x_4=8\\ & x_1+x_2+x_5=7\\ & x_1+2x_2+x_6=12\\ & x_1,\,x_2,\,x_3\ge 0\\ & x_4,\,x_5,\,x_6\ge 0\end{aligned}$$

  - Select (x_4, x_5, x_6) $x_1 = x_2 = x_3 = 0$, so $x_4 = 8, x_5 = 7, x_6 = 12$ $z = 0$
  - Select (x_4, x_1, x_6) $x_5 c = x_2 = x_3 = 0$, so $x_1 = 7, x_4 = 1, x_6 = 5$ $z = 21$
  - Select (x_3, x_1, x_6) $x_5 = x_2 = x_4 = 0$, so $x_3 = 1, x_1 = 7, x_6 = 5$ $z = 23$
  - Select (x_2, x_1, x_3) $x_4 = x_5 = x_6 = 0$, so $x_2 = 5, x_1 = 2, x_6 = 6$ $z = 28$
- Matirx view:
  - $\min -x_1 - x_2 \qquad \rightarrow c + x_1 + x_2 = 0$
    $s.t.\, 2_x 1 + x2 \le 12 \qquad 2x_1 + x_2 + x_3 = 12$
    $x_1 + 2x_2 \le 9 \qquad\quad x_1 + 2x_2 + x_4 = 9$
  - Find pivot: find the column with max non-negative c and calculate division, choose the row with smallest quoticient
  -

	x_1	x_2	x_3	x_4	
x_3	2	1	1	0	12 (6)
x_4	1	2	0	1	9 (9)
c	1	1	0	0	0

	x_1	x_2	x_3	x_4	
x_3	1	1/2	1/2	0	6 (12)
x_4	0	3/2	-1/2	1	3 (2)
c	0	1/2	-1/2	0	-6
    - Select (2, 2) as pivot value and normalize it to 1
	x_1	x_2	x_3	x_4	
x_3	1	0	2/3	-1/3	5
x_4	0	1	-1/3	2/3	2
c	0	0	-1/3	-1/3	-7
- Solving non-linear optimization problems
    - Convex non-linear objective function (NLP), linear or non-linear constraints
    - ilustrated solution technique – Interior Point Methods
    - gradient methods, Newton and Quasi-Newton

Gradient Descent

- Parameters:
    - Learning rate α
    - Parameters: $\theta = (\theta_0, \theta_1, \ldots, \theta_n)$
    - Hypothesis function $h_\theta(x_0, x_1, \ldots, x_n) = \theta_0 x_0 + \ldots + \theta_n x_n = \sum_{i=i}^{n} \theta_i x_i$,where $x_0 = 1$
    - Loss function: $J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{j=0}^{m} (h_\theta(x^{(j)}) - y^{(j)})^2$
- Initialization:
    - $\theta = (\theta_0, \theta_1, \ldots, \theta_n) = (0, 0, \ldots, 0)$
    - $\alpha = 1$
    - ε
- Algorithm:
    - Gradient of $\theta_i$for current loss function: $\frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1, \ldots, \theta_n)$
    - decresing distance $d_i = \alpha \frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1, \ldots, \theta_n)$
    - if $d_i < \varepsilon$, for all i=0,1,..,n algorthm stop, else update θ_i
    - Update parameter: $\theta_i = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1, \ldots, \theta_n)$
- Matrix description:
    - Hypothesis function: $h_\theta(X) = X\theta$
    - Loss function: $J(\theta) = \frac{1}{2}(X\theta - Y)^T (X\theta - Y)$
    - Gradient Descent: $\theta = \theta - \alpha \frac{\partial}{\partial \theta} J(\theta)$

Batch Gradient Descent

- Using all samples to update parameters

Stochastic Gradient Descent

- Using one sample to update parameters

Mini-batch Gradient Descent

- Divide sample into n mini-batch(n=10), and using mini-batch to update the parameters orderly

Simulation modeling

- How to compute the probability distribution of the sum of random variables?
  - $z = x + y$, since there is no PDF and PMF
  - Convolution 卷积: a mathematical operation that allows to derive the distribution of a sum of two random variables from the distributions of the two summands
    $f_z(x) = \int_{-\infty}^{\infty} f_y(z-x) f(x) dx$
- Market campaign simulation modeling