Summary of Chapter 2 : Machine Learning

Yuhang Zhang

Beijing Institute of Technology(BIT)

October 20, 2019

Outline

Outline

Overview

- What to learn?

Overview

- What to learn?
  - Learning denotes changes

Overview

- What to learn?
  - Learning denotes changes
- How to learn? – Key points

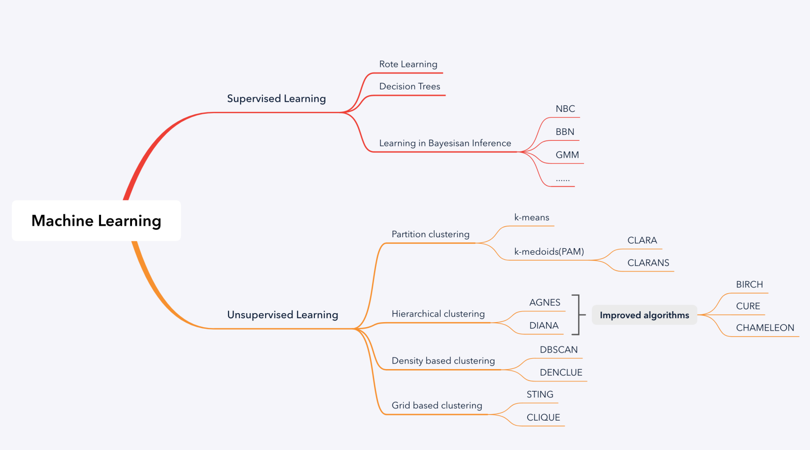
Overview

- What to learn?
  - Learning denotes changes
- How to learn? – Key points
  - Objectiveness

Overview

- What to learn?
    - Learning denotes changes
- How to learn? – Key points
    - Objectiveness
    - Optimization

Overview


Machine Learning
Supervised Learning
Rote Learning
Decision Trees
Learning in Bayesisan Inference
NBC
BBN
GMM
......
Unsupervised Learning
Partition clustering
k-means
k-medoids(PAM)
CLARA
CLARANS
Hierarchical clustering
AGNES
DIANA
Improved algorithms
BIRCH
CURE
CHAMELEON
Density based clustering
DBSCAN
DENCLUE
Grid based clustering
STING
CLIQUE

Outline

What and how to learn?

Learn a function to fit the data

- key points

What and how to learn?

Learn a function to fit the data

- key points
  - Function Structure

What and how to learn?

Learn a function to fit the data

- key points
  - Function Structure
    - Ockham's razor

What and how to learn?

Learn a function to fit the data

- key points
    - Function Structure
        - Ockham's razor
    - Optimize the parameters

What and how to learn?

Learn a function to fit the data

- key points
  - Function Structure
    - Ockham's razor
  - Optimize the parameters
- learning methods

What and how to learn?

Learn a function to fit the data

- key points
  - Function Structure
    - Ockham's razor
  - Optimize the parameters
- learning methods
  - data points : rote learning

What and how to learn?

Learn a function to fit the data

- key points
  - Function Structure
    - Ockham's razor
  - Optimize the parameters
- learning methods
  - data points : rote learning
  - implicit : decision trees, Bayesian Inference

Rote learning

- trade-off between saving states and computing

Rote learning

- trade-off between saving states and computing
- tune the search depth and evaluation function

Elements

Any discrete function

- non-leaf nodes : input variables

Elements

Any discrete function

- non-leaf nodes : input variables
- leaf nodes : output result

Elements

Any discrete function

- non-leaf nodes : input variables
- leaf nodes : output result
- edges : values of variables

Bases of choosing attributes

- Informatino Gain (IG)

$$I(\frac{p}{p+n},\frac{n}{p+n}) = -\frac{p}{p+n}log_2\frac{p}{p+n} - \frac{n}{p+n}log_2\frac{n}{p+n}$$

$$IG(A) = I(\frac{p}{p+n},\frac{n}{p+n}) - remainder(A)$$

Bases of choosing attributes

- Informatino Gain (IG)

$$I(\frac{p}{p+n},\frac{n}{p+n}) = -\frac{p}{p+n}log_2\frac{p}{p+n} - \frac{n}{p+n}log_2\frac{n}{p+n}$$

$$IG(A) = I(\frac{p}{p+n},\frac{n}{p+n}) - remainder(A)$$

- Minimum Description Length (MDL) criterion : $complexity(f(x)) + accuracy(f(x))$

Problem and solution

- overfitting

Problem and solution

- overfitting
- solution

Problem and solution

- overfitting
- solution
  - regularization term

Problem and solution

- overfitting
- solution
  - regularization term
  - post-prune

Bases

- Foundation :

$$P(h|D) = \frac{P(D, h)}{P(D)} = \frac{P(D|h)P(h)}{P(D)}$$

Bases

- Foundation :

$$P(h|D) = \frac{P(D,h)}{P(D)} = \frac{P(D|h)P(h)}{P(D)}$$

  - $P(h)$: Prior probability

Bases

- Foundation :

$$P(h|D) = \frac{P(D,h)}{P(D)} = \frac{P(D|h)P(h)}{P(D)}$$

  - $P(h)$: Prior probability
  - $P(h|D)$: Posterior probability

Bases

- Foundation :

$$P(h|D) = \frac{P(D,h)}{P(D)} = \frac{P(D|h)P(h)}{P(D)}$$

  - $P(h)$: Prior probability
  - $P(h|D)$: Posterior probability
  - $P(D|h)$: Class-conditional probability

Bases

- Foundation :

$$P(h|D) = \frac{P(D,h)}{P(D)} = \frac{P(D|h)P(h)}{P(D)}$$

  - $P(h)$: Prior probability
  - $P(h|D)$: Posterior probability
  - $P(D|h)$: Class-conditional probability

- Criterion :

$$h_{MAP} = argmax_h P(D|h)P(h)$$

Simplified form of class-conditional probability

- Naive Bayesian Classifiers (NBC)

$$P(D|h) = P(d_1, \ldots, d_n|h) = \prod_i P(d_i|h)$$

Simplified form of class-conditional probability

- Naive Bayesian Classifiers (NBC)

$$P(D|h) = P(d_1, \dots, d_n|h) = \prod_i P(d_i|h)$$

- Bayesian Belief Networks (BBN)

$$P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i|P(Parents(x_i)))$$

How to learn?

- fully observable variables : Conditional probability table (CPT)

How to learn?

- fully observable variables : Conditional probability table (CPT)
- only some observable : Multi-layer perceptron

Outline

Overview

Fundamental task :

- Clustering (Similarity Measurement)

Key points :

Overview

Fundamental task :

- Clustering (Similarity Measurement)
- Association Rule Mining

Key points :

Overview

Fundamental task :

- Clustering (Similarity Measurement)
- Association Rule Mining

Key points :

- Similarity measurement

Overview

Fundamental task :

- Clustering (Similarity Measurement)
- Association Rule Mining

Key points :

- Similarity measurement
- Data set division

Overview

Fundamental task :

- Clustering (Similarity Measurement)
- Association Rule Mining

Key points :

- Similarity measurement
- Data set division
- Scalability

Partition clustering

- Blind search

Partition clustering

- Blind search
- Heuristic search

Partition clustering

- Blind search
- Heuristic search
  - k-means algorithm

Partition clustering

- Blind search
- Heuristic search
  - k-means algorithm
  - k-medoids algorithm (PAM)

Partition clustering

- Blind search
- Heuristic search
    - k-means algorithm
    - k-medoids algorithm (PAM)
        - CLARA

Partition clustering

- Blind search
- Heuristic search
    - k-means algorithm
    - k-medoids algorithm (PAM)
        - CLARA
        - CLARANS

k-means algorithm

$$J(X, v) = \sum_{i=1}^{n} \sum_{j=1}^{k} (u_{ik})^m \rho(d(x_i, v_j))$$

$O(tkn)$, local optimality, noises sensitive

k-medoids algorithm (PAM)

- Substitution cost :

$$C_{ih} = \sum_{j} C_{jih}$$

$O(tk(N-k)^2)$, Strong robustness

k-medoids algorithm (PAM)

- Substitution cost :

$$C_{ih} = \sum_j C_{jih}$$

  $O(tk(N-k)^2)$, Strong robustness
- Improved algorithms for large data set:

k-medoids algorithm (PAM)

- Substitution cost :

$$C_{ih} = \sum_{j} C_{jih}$$

$O(tk(N-k)^2)$, Strong robustness

- Improved algorithms for large data set:
  - CLARA : sampling

k-medoids algorithm (PAM)

- Substitution cost :

$$C_{ih} = \sum_j C_{jih}$$

  $O(tk(N-k)^2)$, Strong robustness
- Improved algorithms for large data set:
  - CLARA : sampling
  - CLARANS : repetitive sampling

Hierarchical clustering

Overall methods :

- Bottom-up : AGNES

Problems and Solutions :

Hierarchical clustering

Overall methods :

- Bottom-up : AGNES
- Top-down : DIANA

Problems and Solutions :

Hierarchical clustering

Overall methods :

- Bottom-up : AGNES
- Top-down : DIANA

Problems and Solutions :

- False step sensitive

Hierarchical clustering

Overall methods :

- Bottom-up : AGNES
- Top-down : DIANA

Problems and Solutions :

- False step sensitive
- Improved algorithms

Hierarchical clustering

Overall methods :

- Bottom-up : AGNES
- Top-down : DIANA

Problems and Solutions :

- False step sensitive
- Improved algorithms
  - BIRCH

Hierarchical clustering

Overall methods :

- Bottom-up : AGNES
- Top-down : DIANA

Problems and Solutions :

- False step sensitive
- Improved algorithms
  - BIRCH
  - CURE

Hierarchical clustering

Overall methods :

- Bottom-up : AGNES
- Top-down : DIANA

Problems and Solutions :

- False step sensitive
- Improved algorithms
  - BIRCH
  - CURE
  - CHAMELEON

BIRCH

- Clustering feature (CF) :

$$CF = (N, LS, SS)$$

Additive

BIRCH

- Clustering feature (CF) :

$$CF = (N, LS, SS)$$

Additive

- CF Tree :

B,L,T

BIRCH

- Clustering feature (CF) :

$$CF = (N, LS, SS)$$

Additive

- CF Tree :

B,L,T

- Scales linearly ; order sensitive

CURE

Features :

- AGNES

CURE

Features :

- AGNES
- Representative data

CURE

Features :

- AGNES
- Representative data
- Shrunk toward the mean value

CURE

Features :

- AGNES
- Representative data
- Shrunk toward the mean value
- Random sampling for large scale

CHAMELEON

Dynamic modeling ; Graph based method

- Construct a sparse graph (KNN)

CHAMELEON

Dynamic modeling ; Graph based method

i Construct a sparse graph (KNN)

ii Partition the graph

CHAMELEON

Dynamic modeling ; Graph based method

i Construct a sparse graph (KNN)

ii Partition the graph

iii Merge partitions

Density based clustering

Arbitrary shapes ; Noise handled ; Scan once

- DBSCAN

Density based clustering

Arbitrary shapes ; Noise handled ; Scan once

- DBSCAN
- DENCLUE

DBSCAN

$$N_{Eps}(p) = \{q \in D | dist(p, q) <= Eps\}$$

Parameters : Eps ; MinPts

- Directly density-reachable

DBSCAN

$$N_{Eps}(p) = \{q \in D | dist(p, q) <= Eps\}$$

Parameters : Eps ; MinPts

- Directly density-reachable
- Density-reachable

DBSCAN

$$N_{Eps}(p) = \{q \in D | dist(p, q) <= Eps\}$$

Parameters : Eps ; MinPts

- Directly density-reachable
- Density-reachable
- Density-connected

DENCLUE

- Influence function

DENCLUE

- Influence function
  - Square wave

DENCLUE

- Influence function
  - Square wave
  - Gaussian

DENCLUE

- Influence function
  - Square wave
  - Gaussian
- Density function

DENCLUE

- Influence function
  - Square wave
  - Gaussian
- Density function
- Density attractor X^*

Grid-based clustering

Dense Grid Cell

- STING
  Scale independent ; Non-diagonal boundary

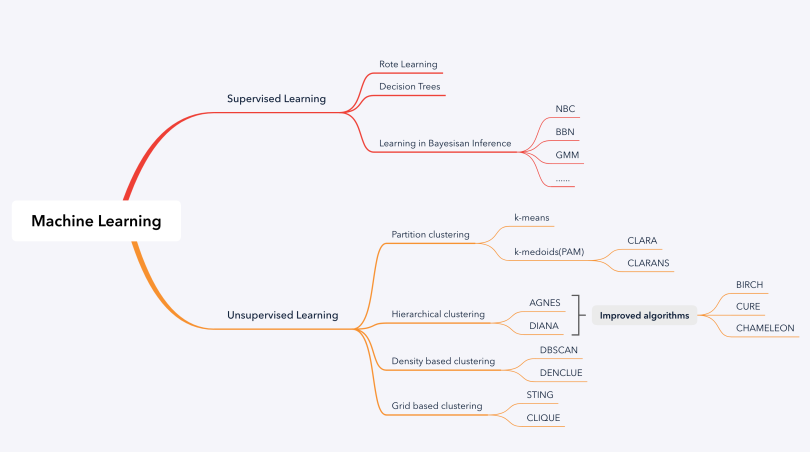
Grid-based clustering

Dense Grid Cell

- STING
  Scale independent ; Non-diagonal boundary
- CLIQUE
  Dimension reduction

Outline

Review


Machine Learning
Supervised Learning
Rote Learning
Decision Trees
Learning in Bayesisan Inference
NBC
BBN
GMM
......
Unsupervised Learning
Partition clustering
k-means
k-medoids(PAM)
CLARA
CLARANS
Hierarchical clustering
AGNES
DIANA
Improved algorithms
BIRCH
CURE
CHAMELEON
Density based clustering
DBSCAN
DENCLUE
Grid based clustering
STING
CLIQUE

Thank you

Thank you for listening!