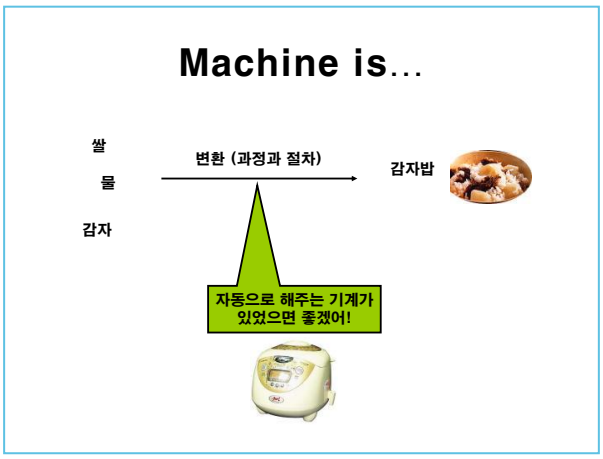
10. 대안알고리즘

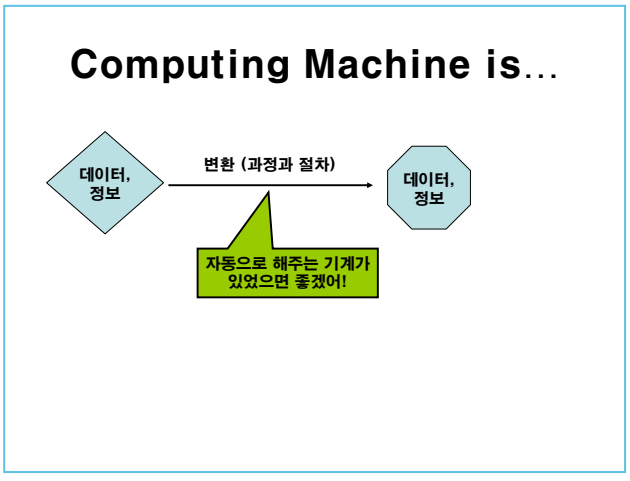
2010 데이터로 표현하는 세상 요약본
고려대학교 김현철 교수
hkim64@gmail.com

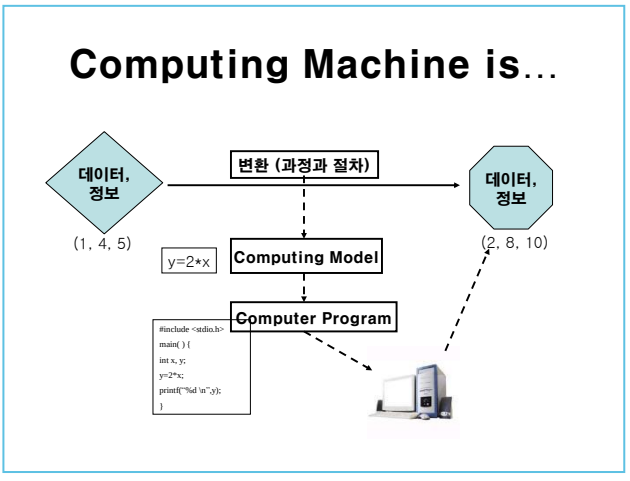
우리는 procedural 알고리즘에 대하여 배웠다. 즉, 문제 해결 과정을 작은 단위로 분해한 후에 그것을 순차적으로 나열하고 그 순서대로 진행하는 알고리즘에 대하여 살펴보았다. 하지만, 이러한 procedural algorithm은 몇몇 문제에 있어서는 unsolvable 하거나 practically impossible한 경우들이 있다. 따라서, 이러한 문제를 해결하기 위하여 procedural algorithm 에 대한 대안적 접근 방법 (alternative approach)가 나오기 시작했으며 그 예는 다음과 같다.

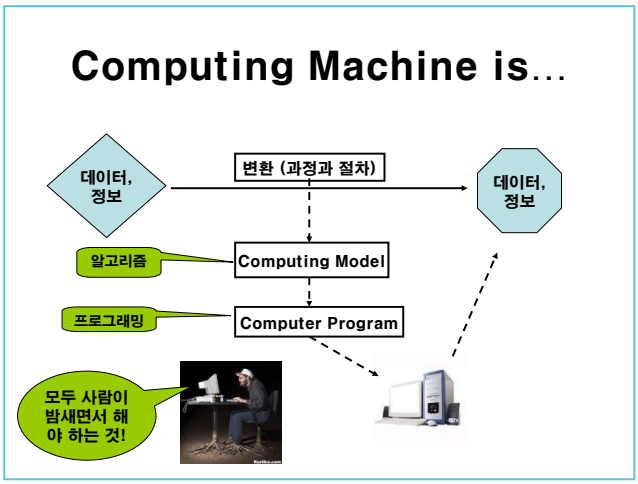
- 인공지능
  - 지식기반(knowledge-based) 방법
  - Data-oriented 방법
    - 기계학습(machine learning)과 data mining
- 인공생명
- 진화 or 유전자 알고리즘
- 뇌 기반 방법, 신경망 알고리즘
- Boinics
- 집단 지성과 휴먼컴퓨팅

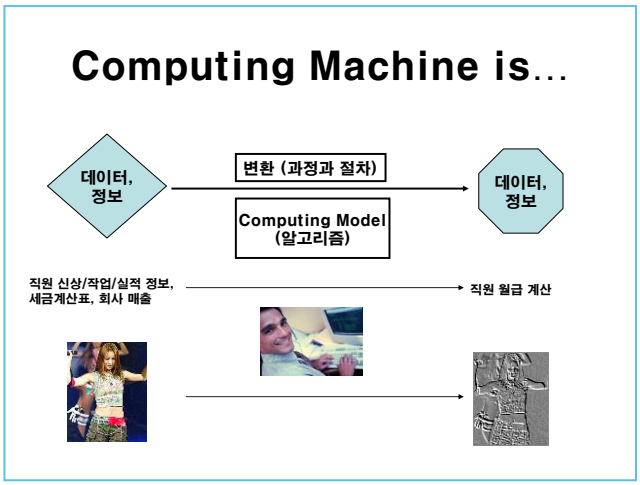
우리는 지금까지 Procedural한 컴퓨팅 모델에 대하여 학습을 하였다. 그 내용들을 다시 한번 검토하여 보자.



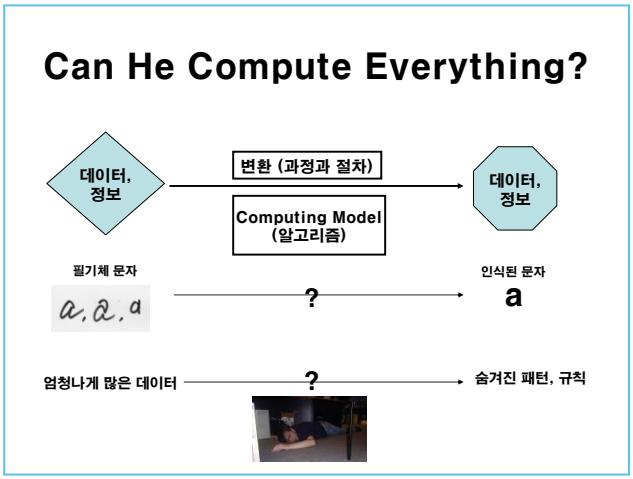


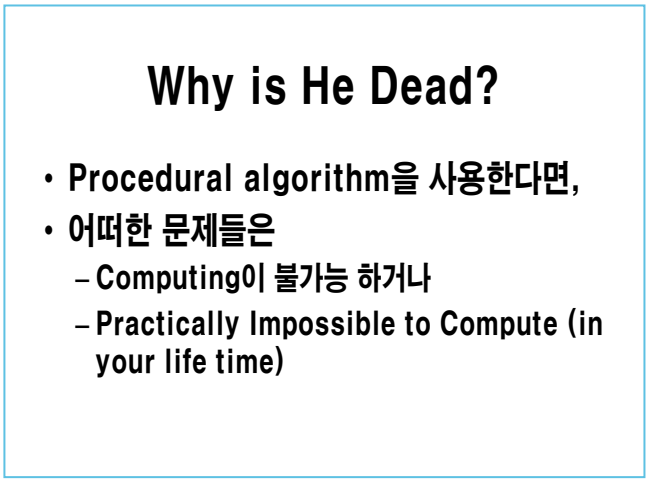






즉, 우리가 생각할 수 있는 문제 해결 방법을 절차적으로 재구성 한 후에, "절차적인 일 처리를 반복적으로 매우 빠르게 처리할 수 있는" 기계가 처리하게 하는 방법으로 우리는 세상의 문제들을 해결하려고 시도 하였다. 하지만, 이러한 절차적 컴퓨팅 방법(procedural computing approach)는 우리의 두뇌의 노동 부하를 감소시켜주고 있는가? 세상의 모든 문제를 해결할 수 있는가?





즉, 반복적 계산으로 처리할 수 있는 것들은 절차적 계산모델로 쉽게 해결할 수 있지만, 그렇지 못한 것들은 어쩔 것인가?

예를 들어서

- ✓ 우리는 쉽게 하고 있는 사람의 얼굴 인식을 절차적 컴퓨팅 모델로?
- ✓ 우리는 쉽게 하고 있는 추론을 절차적 컴퓨팅 모델로?
- ✓ 우리는 쉽게 하고 있는 일반적 문제해결을 컴퓨팅 모델로?
- ✓ 최적화 문제(optimization)문제는? (예: traveling salesman problem)
- ✓ 자연어처리는? 자동번역은?

한글
우리집 강아지는 예쁜강아지 학교갔다 돌아오면 멍 멍 멍 꼬리치며 반갑다고 멍 멍 멍
영어
If my house puppy falls on Hakgyogatda it is Yeppeunganga
bruise bruise bruise that is seductive and glad bruise bruise bruise

이러한 문제들의 처리에는 우리는 인간의 "지능적" 능력이 사용된다고 생각할 수 있다. "지능"이 요구되는 문제들은 어떻게 computing model로 만들 것인가????

지능(intelligence)이란 무엇인가?
"A가 B보다 더 intelligent하다"라고 말할 때, 무엇을 가지고 그렇게 이야기를 할 수 있는가?

"지능"이란 무엇인가에 대한 정의에 대하여 많은 초기 인공지능 연구자들이 고민하였다. 그 중의 두 가지 예에 대하여 설명하도록 하겠다.

1. Turing test
- A test proposed in 1950 by Alan Turing
- A human "A" can communicate with source "B" & "C"
- One source is said to be human and the other is a machine
- "A" must decide which source is human and which is the machine.
- If he can not tell, the machine can be said to be "intelligent".
- Example) 초기 인공지능 프로그램 중 Mycin이라는 환자 진단 프로그램이 있다. 똑 같은 환자 데이터에 대하여 그 프로그램을 적용하여 나온 성능과, 실제 의사 100명에게서 나온 진단 결과를 비교하여, 프로그램의 성능이 의사들의 성능과 비슷하거나 더 우수하면 그 프로그램을 intelligent라고 할 수 있다.

2. Chinese Room
- Chinese Room이라는 방이 있고, 우리는 그 방안에 무엇이 있는지 알지 못한다. 다만 조그만 구멍이 있어서 우리는 그 방안에 있는 사람과 쪽지만 주고 받을 수 있다.
- 쪽지에 영어 단어를 적어서 넣어 주면 잠시 후 가 방안에서는 그 단어의 한자어가 적힌 쪽지가 튀어 나온다.
- 그 Chinese room은 intelligent한가?

이 두 가지 문제는 우리에게 다음과 같은 질문을 던진다.
지능적인 행위를 흉내 내는 것을 "지능"이라고 할 것인가, 아니면 "지능"적인 방법으로 지능 행위를 하는 것을 "지능"이라고 할 것인가.

ex) 비행기, 데이터베이스, Deep Blue, Aibo

Computing Approaches

- **Procedural Algorithms**
  - **Elegant Algorithms to solve general but fundamental problems**
- **Knowledge-Based Approaches**
  - **Use of Knowledge in addition to Data and Information**
  - **Bottleneck in knowledge acquisition**
- **Data-Driven Approaches**
  - **Computing model made of experiences (observed data)**
  - ***Machine Learning***

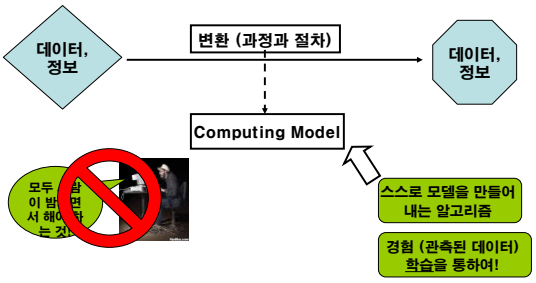
Machine Learning :
Computing Machine that Learns!

- ***Machine Learning is the study of computer algorithms that improve automatically through experience.***
  –Machine Learning, Tom Mitchell, McGraw Hill, 1997-

Machine Learning :
Computing Machine that Learns!

- ***Machine Learning** is the study of computer algorithms that improve automatically through experience.*
  –Machine Learning, Tom Mitchell, McGraw Hill, 1997-

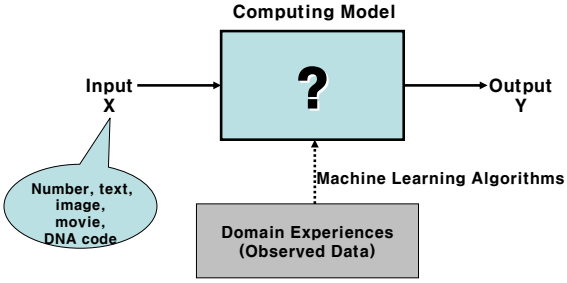
Computing Machine that Learns…



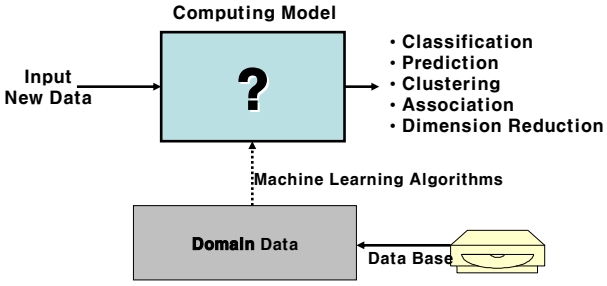
Machine Learning Algorithms

- Data-driven approach
- Non-parametric approach
- Human flavor to knowledge representation (structure)
- Can deal with missing and noisy data, and uncertainty

Abstraction of ML-based Computing



Abstraction of ML-based Computing



Data Mining

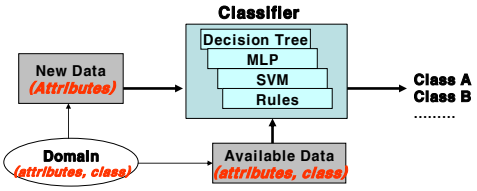
- **Efficient Extraction of implicit, previously unknown, and potentially useful information/patterns from data**

ML/DM Strategies

- Classification
- Clustering
- Association
- Etc.

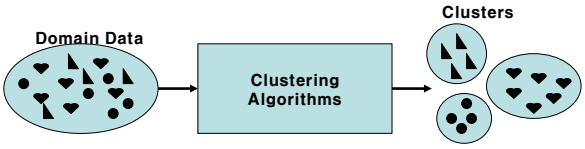
Classification

- Decision Tree
- Neural Networks (MLP)
- Support Vector Machine
- Rule Induction



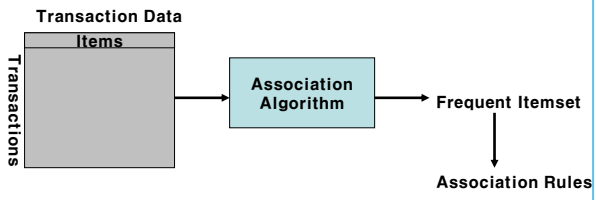
Clustering

- A way to segment data into groups that are not previously defined.
- Distance-based Clustering
  - Euclidean Distance
  - Non-Euclidean Distance
- Probability-based Clustering



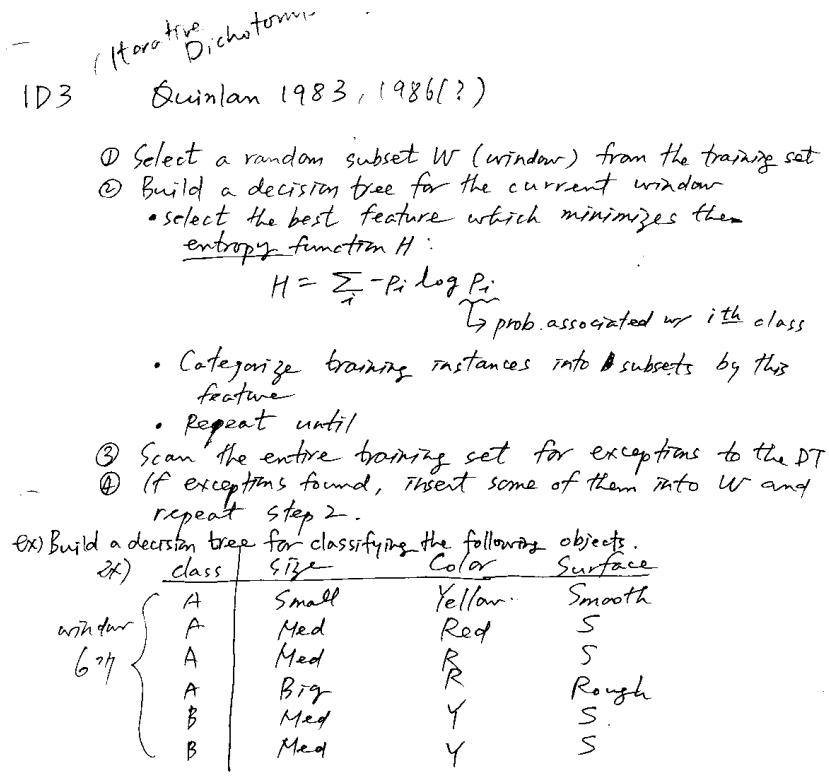
Association

- Comes from *Market Basket Analysis*
- Frequent Itemset
- Association Rule



결정트리

Decision tree 알고리즘 중 가장 오래된 것은 아마도 1983년의 ID3 일 것이다. 그 이후에 나온 많은 알고리즘도 모두 이 ID3를 기반으로 해서 조금씩 변형 된 것이다.
아래의 알고리즘과 예를 먼저 살펴보도록 하자.



예를 먼저 보도록 하자.

3개의 feature (즉 attribute) size, color, surface를 가지고 그 과일 instance의 class를 결정짓는 classification model을 만들고자 한다. Size는 3개의 값, color는 2개의 값, surface는 2개의 값을 가지고 있으므로 모두 3*2*2=12개의 possible combination 이 가능하다. 이것이 이 도메인의 전체 크기이다. Training data로는 위의 6개가 주어졌고, 이 6개의 instance를 가지고 분류모델을 만든다는 것이다.

결정트리를 만들 때에는 가장 먼저 트리의 루트노드를 무엇으로 할 것인가가 중요한 문제이다. 당신이 생각하기에, 위의 3가지 attribute중에서 어느 것이 가장 중요한 (즉, 어느 것이 클래스 A,B로 분류하는데 좋은) attribute일 것이라고 생각하는가? 찬찬히 보면서 생각해 보라. Color를 가지고 하면, red이면 모두 A 클래스, yellow이면 거의 B 클래스가 된다. 흠… surface를 보면 smooth일 때에는 거의 반반으로 나뉘어 지기 때문에 그리 도움이 될 것 같지 않다.

이렇게 각 feature들에 대해서 그 feature가 classification 에 얼마나 도움이 되는가를 정량적으로 계산해 주는 식이 필요할 것이다. 그래야 그것으로 어느 것을 node로 정할 것인지를 효과적으로 구할 수 있을 것이다. 위의 ID3 알고리즘에서는 각 feature들의 entropy를 계산하였다.

이 entropy를 설명하기 전에, 먼저 decision tree의 기본 아이디어를 우리들이 잘 알고 있는 "스무고개" 게임을 통해서 설명하도록 하겠다.

스무고개 게임

이 게임은 한 사람이 어느 물건을 마음 속으로만 생각하고 있으면, 다른 사람이 스무 번의 yes/no 질문을 순차적으로 하여 상대방이 생각하고 있는 그 물건이 무엇인지를 알아 맞히는 게임이다. 이 게임에서의 domain은 무척 포괄적이다. 이 세상의 모든 것이 이 domain에 들어간다. 질문은 yes/no로 대답할 수 있는 것만 가능하므로, 하나의 질문으로 도메인을 이등분 할 수가 있다. 따라서 그 범위를 좁혀 나가기 위하여서는 질문이 매우 효율적이어야 한다. 효율적이라는 말은 그 질문에 의하여 도메인의 공간

을 최대한 많이 제거할 수 있어야 한다는 것이다.

예를 들어서, 우리가 흔히 하는 첫 번째 질문은 "생명이 있는 것입니까?"이다. yes라면 무생물쪽은 모두 탐색공간에서 제거시켜 버릴 수가 있고, no라면 유생물 쪽은 모두 고려대상에서 제거시켜 버릴 수가 있다. 즉, 효율적인 질문이라는 것은 그 질문에 의하여 도메인 공간을 비슷한 크기의 두 개의 공간으로 이등분 시킬 수 잇는 질문이다. 만약에 질문이 "검정색 볼펜입니까?"라고 물으면, 그리고 대답이 no였다면 그 큰 도메인 공간에서 단 하나의 item만 제외한 나머지 n-1의 공간을 가지고 다시 탐색을 해야 하는 것이다. 귀중한 질문을 하나 낭비한 샘이다.

만약에 질문이 정확하게 탐색공간을 2등분할 수 있다고 한다면, 20번의 질문에 의하여 우리는 전체 공간을 2^{20}=1,048,576 즉 백만개의 같은 크기의 조각으로 나눌 수 있다는 것이다. 만약에 1백만 개의 물건들 중에서 한가지를 생각하고 게임들 시작하고, 질문은 정확하게 남은 탐색공간을 이등분하는 것이라고 한다면, 그 백만개 중의 하나의 물건을 반드시 찾는 다는 것을 보장한다.

우리 아이(6살)과 하는 게임을 이야기 해 보이도록 하겠다.

1에서 20까지 숫자 중에서 하나 생각해봐. 생각했어요.

10보다 크니? 네.

15이상이니? 아니요.

13이상이니? 아니요 (흠. 그러면 11,12,13중의 하나일 것)

그럼.. 12보다 크니? (약간 놀라는 표정으로) 아니요. (그럼 11 아니면 12만 남았음)

그럼 12니? (화들짝 놀라며) 네!

이렇게 5번 만에 맞힐 수가 있다. 이것은 log_2(20) = 4.*** 이 되므로 영역을 2등분 시켜나간다면 최대 5번안에 맞힌다는 것을 보장한다. 10번의 질문을 할 수 있다고 한다면 2^{10}=1024이므로 아이에게 1에서 1024까지의 숫자 중에서 아무거나 하나 생각하라고 할 수가 있다. 스무고개는 1,048,576개를 커버할 수 있다. Log_2(1,048,576)= log_2(2^{20})= 20.

이렇듯, 질문은 탐색공간을 최대한 줄여 나갈 수 있는 효율적인 것이어야 한다. 결정트리는 이러한 스무고개와 비슷한 것이며, training data를 봐서 가장 효율적인 질문이 무엇이어야 하는가를 결정할 수 있어야 한다. ID3에서 사용한 그 기준, 즉 entropy에 대하여 살펴 보도록 하자.

이제 entropy에 대하여 알아 보았으니, 다시 과일데이터로 돌아가보자.

Entropy 의 계산은 다음과 같다.

Entropy(p1, p2,…, pn) = -p1*log_p1 -p2*log_p2-…….-pn*log_pn.

예를 들어,

Size=med 의 entropy를 계산해보자. Size=med에 의하여 해당하는 데이터 4개는 2개는 클래스 A로, 다른 2개는 클래스 B로 나뉘어진다. Size=med라는 정보의 entropy는

Entropy([2,2]) = -(2/4)*log_(2/4) - (2/4)*log_(2/4) = 1/2+1/2 = 1bit.

자 그러면,

size라는 attribute의 entropy는 어떻게 구할 것인가? size라는 attribute는 3가지의 value를 가진다, 즉 small, med 그리고 big. 각각의 entropy를 구하고 해당되는 instance의 수에 비례한 확률적 weight값으로 평균을 구하면 된다.

이 과정이 끝난 후에, training window 밖에 있는 다른 training set으로 테스트 하여 exception을 찾아서 tree를 만드는 과정을 다시 한다.

Build a decision tree for classifying the following objects.

예)

	class	Size	Color	Surface
with four 6개	A	Small	Yellow	Smooth
	A	Med	Red	S
	A	Med	R	S
	A	Big	R	Rough
	B	Med	Y	S
	B	Med	Y	S

각 attribute의 entropy 계산

Size H : $= \frac{4}{6}\left(-\frac{1}{1}\log\frac{1}{1}\right) + \frac{4}{6}\left(-\frac{2}{4}\log\frac{2}{4} - \frac{2}{4}\log\frac{2}{4}\right)$ (Small, Med(A), Med(B))

전체 6개 중에 S 1, M 4, B 1

$+ \frac{1}{6}\left(-\frac{1}{1}\log\frac{1}{1}\right) = 0.462$ (Big)

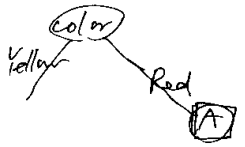
Color H : $= \frac{3}{6}\left(-\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}\right) + \frac{3}{6}\left(-\frac{3}{3}\log\frac{3}{3}\right)$ (Yellow, Red 모두 A)

$= 0.318$

Surface H = 0.56

따라서 Color의 entropy가 minimum!

Color = Red 의 경우에 entropy가 0 이다 즉 모두 A.



to make further distinctions, another attribute 를 사용해야 함

Color = Yellow 의 경우에 다시 entropy 계산,
즉 training set (이 feature의 경우) 은

	Size	Color	Surface
A	S	Y	S
B	M	Y	S
B	M	Y	S

이므로

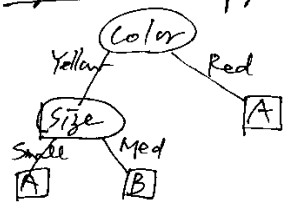
Size $H = \frac{1}{3}\left(-\frac{1}{1}\log\frac{1}{1}\right) + \frac{2}{3}\left(-\frac{2}{2}\log\frac{2}{2}\right)$ (Size S, Size M)

$= 0$

Surface $H = \frac{3}{3}\left(-\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}\right)$ (Surface S)

$= 0.636$

따라서 Size의 entropy가 minimum!



//

ID3 selects the features which minimize the entropy function and thus best discriminates among the training instances.
ID3의 문제점
- 모든 attribute values들이 다 있어야 한다. No missing values.
- 값들은 discrete (즉, nominal or categorical) 값이어야 한다.
- Domain size와 dimension에 따라서 tree의 size가 엄청나게 크게 증가할 수 있다.

Enhancement from ID3
- CART, C4.5, C5.0 etc.

클러스터링

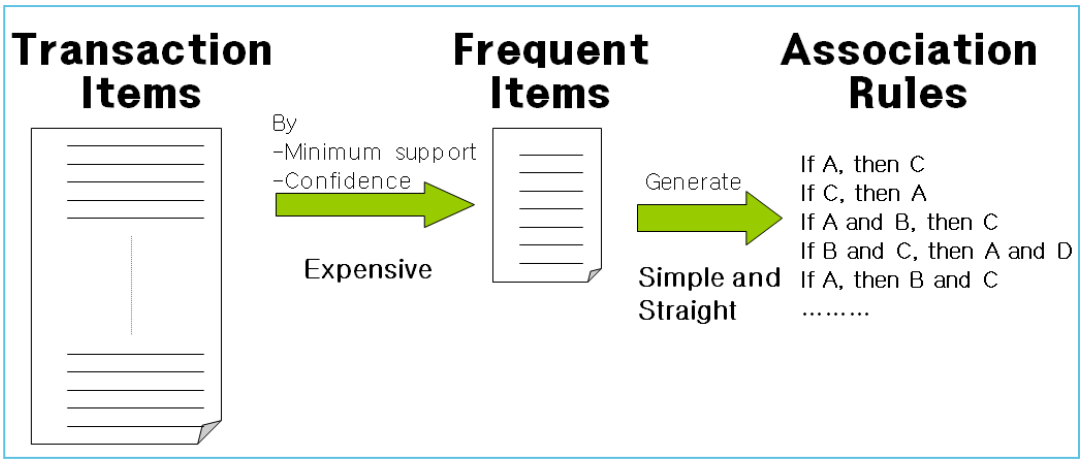
K-Means 알고리즘

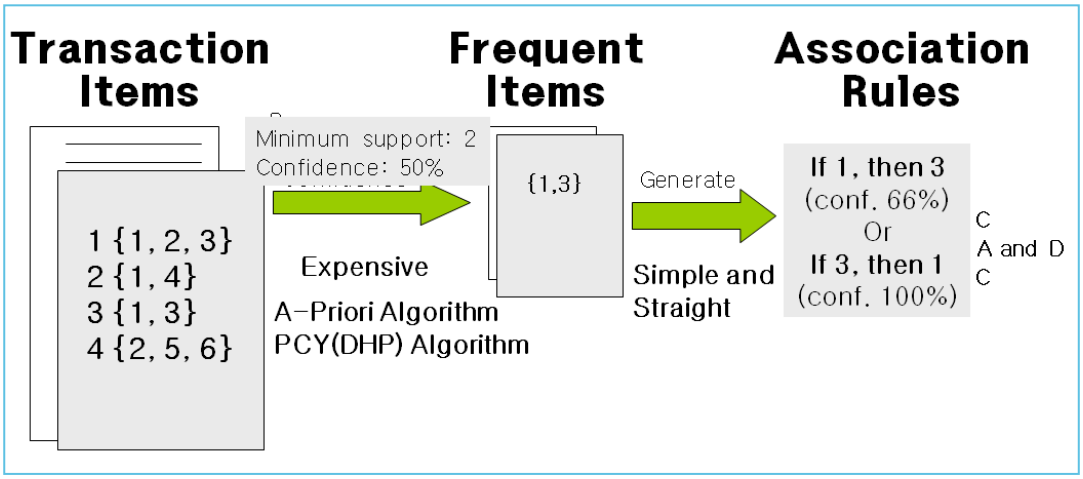
- 이 알고리즘은 real-valued 데이터에서만 사용할 수 있다. 만약a categorical attribute가 있다면 그 attributes를 빼고 사용하거나 혹은 그것을 수치 값으로 변환하여 사용하여야 한다. 예를 들어, color라는 attribute에 red, blue, green, brown의 값이 있다고 하자. 한가지 방법은 임의의 수치 값을 각 color에 지정하여 주는 것인데 여기에도 여러 가지 단점들이 있다.
- 우리는 만들어지게 되는 클러스터의 개수에 해당하는 값을 먼저 선택하여야 하는데, 만약 우리가 잘못된 선택을 하게 된다면 이것 또한 문제가 된다. 이 이슈를 해결할 한가지 방법은 서로 다른 K 값으로 여러 번 알고리즘을 돌려보는 것인데, 이렇게 하여 이 데이터에는 몇 개의 클러스터가 있을 것인지에 대한 "감"을 잡을 수 있게 될 것이다.
- K-Means 알고리즘은 데이터에 있는 클러스터들이 비슷한 크기일 때 가장 성능이 좋다. 이 경우에서는 만약 optimal solution 이 다른 크기의 클러스터로 표현된다면 K-Means 알고리즘은 가장 좋은 solution 을 찾지 못할 가능성이 높다.
- 클러스터가 만들어지는데 중요한 attributes 가 어느 것인지 설명할 수 있는 방법이 없다. 이런 이유로 관련도 없는 attributes 들이 들어가게 되어서 최적(optimal)결과를 만들어 내는 것을 방해할 수 있다.
- 만들어진 클러스트의 특성에 대한 설명이나 해석을 하기가 힘들기 때문에 무엇이 발견되었는지에 대한 해석은 우리의 몫이 되어 버린다. 하지만 supervised 데이터마이닝 도구를 사용하여 unsupervised clustering algorithms 으로 만들어진 클러스터의 특성에 대한 파악을 해볼 수는 있다.

연관규칙

〈장바구니 분석〉

개요
- 문제정의
  - Transactional data
  - Support, confidence
- Application: Market basket Analysis 장바구니 분석
- Complexity
- Interestingness 흥미도





If 기저귀, then 맥주

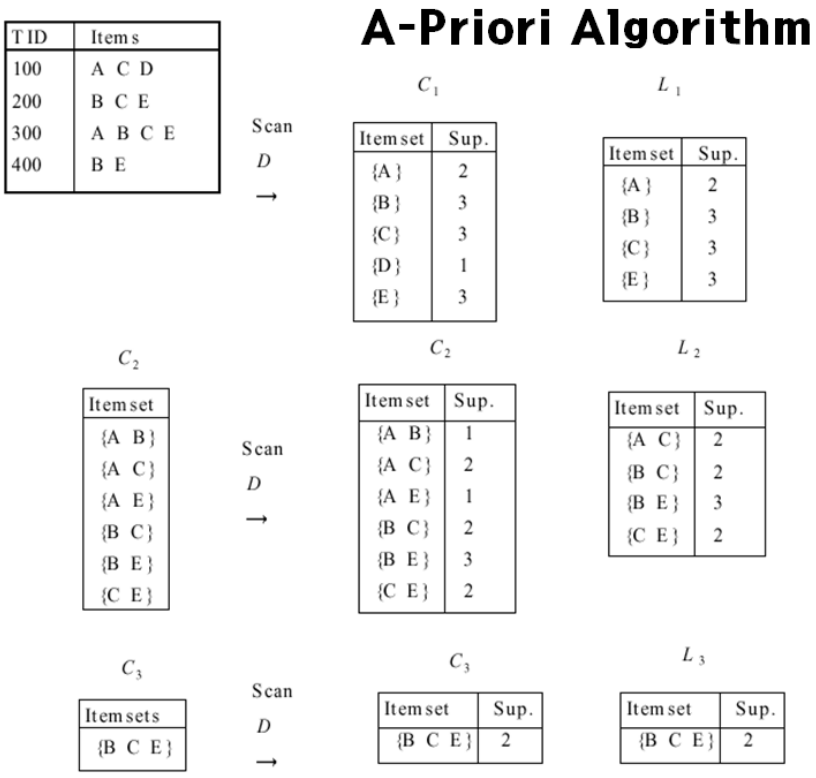
전략:

1. 기저귀와 맥주를 함께 진열/포장
2. 기저귀와 맥주를 멀리 따로 진열
3. 기저귀 값 인하, 맥주 값 인상

If 맥주, then 기저귀 ??

Complexity 문제

- Size2인 itemset을 찾는 complexity는?
- Size3은?
- Size4는?



Applications

Market Basket Analysis가 제일 많이 사용하고 있지만 그 외에도 여러 가지 어플리케이션에 사용할 수가 있다.

예를 들어,

- Basket=documents; items=word인 경우는?
- Basket=web pages; items=link인 경우는? Frequent items maybe pages about the same topic.

Interestingness

- 찾아진 an itemset의 item들은 정말 associated되어 있는가?
- 우연히 associated된 것은 아닌가?
- 아무 의미 없는 rule은 아닌가?
- 어떻게 판별할 것인가.

예를 들어〉 아까 본 사례인 〈기저기와 맥주〉 정말 둘 간의 〈의미적인〉 association이 있는가? 아니면, 단지 그냥 같이 팔린다는 것인가.

누구나 구입하는 item은 찾아진 연관규칙에 모두 나타난다.

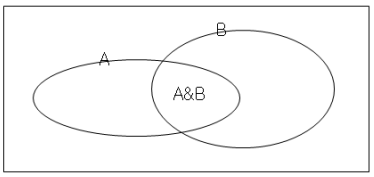
이것을 item간의 연관성이라고 판단할 수 있는가. Support와 confidence로 판단 불가능.

Interestingness를 적용하자.

- Correlation/lift
- Conviction

Measures for rule (A->B)

- Support(지지도) : P(A, B)
- Confidence(신뢰도) : P(B|A)



More measures for A->B

- Correlation (Lift)
  - P(A, B)/(P(A)*P(B))
  - A와 B가 독립적인 사건일 경우 P(A,B) = P(A)*P(B)
  - Lift=1 tells 서로 독립적
  - Lift>1 tells 정적(positive) 연관성
  - Lift<1 tells 부적(negative) 연관성

More measures for A->B

- Conviction (확신도)
  - P(A)*P(~B)/P(A, ~B)
  - Logic에서 A->B 은 (~A or B)와 동치
    - A->B
    - ~A or B
    - ~(A, ~B)
    - ~[P(A,~B)/(P(A)*P(~B))
    - P(A)*P(~B)/P(A, ~B)

활용사례

장바구니 분석에서 사용된 연관규칙 추출방법이 적용될 수 있는 사례를 찾아서 문제 정의를 해보자.

Ex) 건강검진 데이터, 수능 데이터

.끝.