Data mining & Analysis

Ensemble Methods

특징

- 각 classfier 완전히 똑같으면 안됨
- classifier 독립적이어야함
- error rate 구하는 법 , error rate는 절반 이상이 틀리지 않을때 낮아짐
  - e < 0.5

constructing a ensemble

- manipulating the training set
  - bagging, boosting
- manipulating input features
  - random forest
- manipulating algorithms
  - decision tree, ANN

Voting approaches

- simple majority voting
- weighted majority voting

Bias and Variance

- High Bias → 유연성 👎, 에러 ☝ → Underfitting
- High Variance → 유연성 좋음, 오버피팅위험 → Overfitting
  - combining multiple classifiers → reduce the overall variance

Bagging

- Bootstrap aggregating

특징

- 뽑을때 random 하게
- 원본과 같은 n개 뽑아야함(same size)
- 같은 확률분포 사용해야함(same probability)
- replacement 허용 → 뽑힌애들 돌려놔야함
- 뽑힌 애들 별로 round 표수 측정 → 다수결 선택

- High variance 선택해서 variance 줄이는 방향으로, High bias may not improve
- 독립적으로 모델 만들어짐

Boosting

- weight를 줌, 앞에 애들이 틀린거에 weight를 더 많이 줌
- 하나의 모델만(앞의 모델 참고해 모델 생성) → sample 생성시에 배깅 사용가능, 처음에는 equal weigts
- 틀리는 놈에 집중해 두번째 모델 생성
  - Update the weight
    - Incorrect → Increase
    - Correct → Decrease
- 반복반복
- error rate 낮으면 → Importance(alpha) 높아짐
- error rate 높으면 → Importance(alpha) 낮아짐
- 그냥 투표하면 안되고 importance 곱해주고 비교해야함

Random Forests

- decorrelated decision trees
- Bagging과 같은 기법을 사용함(same size/same probability/replacement)
  - randomly selected
  - but, bagging만 쓰면 안됨
- p개 랜덤하게 봅고, 그 중 best information(maximum gain가지는) 선택
  - splitting attribute 모두 달라짐
- every leaf가 pure해질때 까지 반복
- majority vote
- Unpruned trees → low bias, high variance
- independently sampled dataset
- different subset
- 장점
  - variance높은 tree 여러개 만듦, attribute 여러개 선택
    - strong and decorrelated and not overfitting
  - robust to overfitting
  - fast and robust even in high-dimensional setting
- Hyper parameter : p

- small value of p
  - tree간 다양성 커짐, bias 커짐
  - 극소수 → tree각각의 파워 낮아짐
- large value of p
  - 트리 비슷해짐, 거의 모든 attribute선택, correlation
  - 각각의 tree 파워풀해짐

Random Forests vs Bagging

- 배깅
  - strong point가 항상 split point로 사용됨 → tree 비슷해짐
  - tree간 상호관계큼
- random forest
  - best split point 찾는것 뿐만 아니라 p개 랜덤 선택 그 중 best 선택 → 상호관계 떨어짐
  - 다양성 보존됨
  - decorrelated

Multiclass Problem

- 기존 binary → binary 여러개 모아서 multi class로 사용가능

1. One-Against-Rest
   a. 각각에 대해 k개 binary classifier 생성
   b. 가장 confidence, probability 높은애 선택
2. One-Against-One
   a. K(K-1)/2 binary classifer
   b. k(k-1)/2개 binary classifier 생성 → 모든 페어(중복제외)에 대한
   c. 가장 나은거 선택 및 판정

Association Analysis

- Frequent itemsets
- association rules
- Two keys
  - 계산비용 엄청남

    - 열심히 분석해도 의미없을 수 있음
- itemset, transaction
- support count → 카운트한것
- support → 비율, 실생활에서 더 중요 > minsup
    - 우연히 발생한 걸수도 있다
    - 너무 수치가 적으면 의미 없음
- confidence
    - x 산사람이 모두 y삼 의미 X
    - X 산사람 몇 %가 Y를 삼
    - 원인이 아니다!
- Association Rule
    - X → Y : x, y 둘다사고, x 산사람이 y까지 삼
    - s(X → Y) : (XUY)/N 비율로 구함 > minsup
    - c(X → Y) : (XUY)/X → X 산사람중 X&Y같이 삼 > minconf
- Brute-Force approach
    - every possible rule 너무 많음 → 비용 너무 큼
    - [Diaper, Milk]가 min support 못넘기면 [Diaper] → [Milk]도 min support 못넘김
    - 여전히 비용 비쌈
- Improve approach
    - Frequent itemset 찾음
    - 그러면 candidate rule은 모두 min support를 넘김 → minimum confidence 체크해야함
    - 비용 비쌈

Apriori

- candidate itemset이 너무 많음
    - 어떤 itemset이 frequent하면, subset도 frequent 할 것임
    - 어떤 집합이 frequent하지 않으면 걔를 포함한 superset도 frequent하지 않음
- finde 1 frequent itemset → 2 frequent itemset → 반복
- candidate procedure
    - complete → 모든 후보가 빠지지않고 나와야
    - non-rebundant → 중복이 일어나지 않아야
- F(K-1) * F(K-1) Method
    - 알파벳 sorting하고

    - 앞에 (k-1)-itemset 겹치는애
    - complete and no duplicate
- 계산비용
    - support threshold 증가 → 비용 감소
        - 감소 → 후보 많아짐 → 비용 증가
    - number of items 증가 → 비용 증가
    - number of transactions → 비용 증가
    - average transaction width → 비용 증가
- maximal frequent itemsets를 찾아야함

FP-Growth Algorithm

- Apriori보다 월등히 빠름
    - apriori : 후보 만들고 스캔 → 스캔 → 디스크비용 높음
- radically different approach
    - 메모리에 트리 올려놓음 → 훨씬 빨라짐 → but, 알고리즘 어려움
    - 압축된 데이터
        - small enough to fit into main memory
- step
    - scan count each item
    - discard infrequent item
    - sort item
    - extend fp-tree
- original dataset보다 사이즈 작을수록 좋음
    - share되는 item 많을수록
    - Best : same item 많음
    - Worst : none of the transactions → 서로 겹치지 않음
- support 높은애 → 여러군대 나타남 → 겹치는거 많음 → common → 앞으로 보내야
- 앞에 low support item → tree 커짐
- pointer 커넥팅하면 그 아이템 몇번 나왔는지 알 수 있음
- 특징
    - 중복된 itemset 존재하지 않음
    - FP-growth depends on the compaction factor
        - 압축이 덜되면 very bush → 느림
        - 압축이 잘되면 shallow → good

Evaluation of Association Rules

- 흥미롭지 않을 수 있음 ex) {butter} → {bread}
- qualiti 측정해야함 객관적인 measure 바탕으로
- interest factor or the lift!
  - I(X→ Y) = (X→Y)/(X) / (Y)/N = C(X→Y)/(Y)/N
  - I(X→Y) = 1 → X has no influence on Y
  - I(X→Y) < 1 → X discourages Y
  - I(X→Y) > 1 → X positively affects Y

Association Analysis → sequential patterns

- temporal information
- event-based data have sequential nature
- ex) <{bread}, {diaper}>
- Sequence
  - ordered list of elements(엘레멘트의 리스트)
- Element
  - 여러개의 event로 구성
- k-sequence
  - k개의 event(≠ element)로 이루어진 sequence
- subsequence s > sequnce t에 포함
  - t is contained s
- min support 이상인 sequence를 찾아야함
  - sequential pattern 모두 찾기 → substantially larger
  - {i1, i2} & {i2, i1} → same item
  - <{i1}{i2}> & <{i2}{i1}> → different item → possible sequence infinite
  - Apriori principle 사용가능
    - <{a}{b}> frequent하면 <{a}> & <{b}>도 frequent
    - k-1 sequence 사용
    - apriori-like algorithm
    - k-1개 merging 맨앞 event, 맨뒤 event 제거하고 겹치는 부분 합침
      - <{1}{2}{3}> + <{2}{3}{4}> = <{1}{2}{3}{4}>
      - <{1}{5}{3}> + <{5}{3,4}> = <{1}{5}{3,4}>

        - 자기랑 자기 조합 가능
        - element안 event sorting 해야함, element를 sorting하면 안됨!
    - merging procedure is complete → 하나도 빼놓지 않고 frequent한 애들 후보로둠
    - non rebundant → 조합을 만드는 애 딱한가지 방법만 존재함
    - 앞 조합이 infrequent하면 superset도 infrequent

Cluster

- purpose
    - understanding
    - utility
        - summarization
        - compression
        - Efficeintly finding nearest neighbors
- Goal
    - similiar to one another → maximize similarity within a group
    - different from the objects in other gorups → maximize difference between groups
- Classification vs Clustering
    - Classficiation → supervised classficiation
    - Clustering → unsupervised classification
- Segmentation, Partitioning vs Clustering
- Clustering Types
    - Partitional clustering
        - non-overlapping(겹치지 않는 하나의 클러스터)
    - Hierarchcial clustering
        - 중복허용 → 계층적으로
    - Exclusive clustering
        - single cluster로 칼같이 나눔
    - Overlapping(or non-exclusive) clustering
        - 중복 허용
    - Fuzzy clustering
        - 모든 클러스터에 속할 확률 다 찍어줌
    - Complete clustering
        - 데이터 빼놓지 않고 모두 집어 넣음

    - Partial clustering
        - 필요없는건 날려버림
        - BIRCH, CURE 등 알고리즘
    - Prototype-based
        - prototype(centroid)기준으로 거리가까운애 모음
        - ex) k-means
    - Density-based
        - 밀집된 공간 → 클러스터
    - Graph-based
        - 약한 부분 cutting 가능

K-means

- prototype-based, partitional clustering
- step
    - 아무데나 k점 찍음
    - 데이터 모음
    - centroid 업데이트
    - 데이터 모음
    - 중심점 업데이트
    - 반복 → 멈출때까지
- always converges → 계속 돌리면 언젠가 centroid 확정됨
    - 장) 안정적
    - 단) 중심 잘못잡으면 local minimum(≠ global minimum) 수렴
- Centroid 측정 방법(a proximity measure and an objective function)
    - Euclidean distance
        - objective function → minimize SSE(squared error)
        - centroid = mean
    - Cosine similarity
        - objective function → maximize the cohesion
        - centroid = mean
    - Manhattan distance
        - objective function → minimize dist(c,x) 제곱존재x
        - centroid = median
- 초기값 위치만으로 좋을지 나쁠지 알 수 없음
    - 제일 좋은 결과 → SSE 최소화

- 랜덤하게 여러번 돌려도 결과 나쁠 수 있음
- 초기값 정하는 방법
  - Pre-clustering(사전 클러스터링)
    - take sample of points
      - 샘플 너무 많이뽑으면 → 시간 오래걸림
      - K < sample size
    - hierarchical clustering
  - Selecting the fathest point(가장 먼점 선택)
    - outlier 피해야함
    - 먼점 찾기위한 cost 줄여야함
    - K-means ++
      - farthest point 잡는 기법
      - d(x)^2 거리의 제곱에 비례하는 probability 사용
- K-means 단점
  - different size 구분하지 못함
  - different desity 구분하지 못함
  - non-shperical shape → 구형이 아닌 형태의 클러스터링 못함(원형에 적합)
  - outlier에 취약
- 장점
  - simple & wide variety of data types
  - multiple runs

Agglomerative Hierarchical Clustering

- hierarchical clustering(prototype-based or graph-based)
- Agglomerative vs Divisive
  - agglomerative → 가장 가까운 점 뭉쳐나감
  - divisive → 먼점 끼리 뭉쳐감
- Dendrogram
  - relationship
  - order 보여줌
- Defining Proximity(linkage function)
  - MIN (single link or single linkage)
    - closest two points
    - non-elliptical(원형이 아닌) shape에 효과적
    - noise, outlier에 취약함

    - MAX(complete link or complete linkage)
        - farthest two points
        - 너무 먼점 묶이지 않음
        - 원형 추구, 적당한 규모 유지하려함
        - noise, outlier에 어느정도 유지함
    - Group average(average link or average linkage)
        - average → max 에 가까움
        - globular clusters
- Ward’s method
    - 두 클러스터 결합시 반드시 SSE 증가 → 이 증가양을 감소시킴 → “WARD”
    - K-means 클러스터링과 동일한 objective function
    - effective method for noisy data
    - similar to the group average
    - centroid 구할 수 있는 경우에만 사용가능
        - ex) {1,2,3} (ok) {가,나,다} (x)
- centroid method → inversion이 일어날 수 있음
    - 뒤로 갈수록 먼애들끼리 결합해야하는데 짧은애랑 결합함
- key issue
    - 초기값 중요하지 않음
    - 이론적 근거 부족 → 객관적 global objective function 부족
        - locaaly
        - 복잡한 최적화 피해야함
    - 컴퓨터 비용 비쌈
    - singleton, small cluster알아서 merge하지 않음
        - outlier 자동적 추출가능
        - 객관적 기준은 없음 → 사람이 판단해야

DBSCAN

- density-based, partial clustering
- 사람의 생각 = a region of high density = a cluster
- Center-based approach
    - 자기자신 포함해 반경(EPS) 안 점 세기
- problems
    - 반경이 너무 크면 → 모든 점이 비슷한 큰 스코어
    - 반경이 너무 작으면 → 점 별로 없음, 밀도 낮음

- classfication of points
  - core points
    - 내 주변 밀도가 높고, 클러스터 내부
    - MinPts 넘고, EPS 안에 있어야함
  - Border points
    - 코어포인트 아님
    - 가까운 거리 안에 코어포인트 존재
  - Noise points
    - 코어포인트 x, 보더포인트 x
    - 클러스터링에 포함하지 않음
    - minpts 못넘음
- 기법
  - core 모아서 클러스터 만듦
  - border point 연결
  - noise 버림
- k-dist
  - k-dist가 작으면 k-dist < eps
    - 클러스터 내 포함
    - core points
  - k-dist가 크면 k-dist > eps
    - 클러스터 밖
    - noise points, border points
  - k가 너무 작으면
    - noise나 outlier도 클러스터에 포함
  - k가 너무 크면
    - small cluster도 noise에 포함
  - DBSCAN 알고리즘은 K=4 사용
- DBSCAN은 구역별로 density 달라지면 구별 힘듦
- 장점
  - 노이즈에 강함
  - 다양한/임의의 모양 잘찾음
  - k-means에서 발견하지 못했던 다양한 클러스터 찾을 수 잇음
- 단점
  - 밀도가 구역마다 달라지면 문제 생김

    - 고차원 데이터에 문제 있음 → 모든 클러스터링 기법에 대한 문제이기도
    - 각 점마다 밀도 계산 → 컴퓨터 비용 많이 소요

Anomaly Detection

- outlier
- unusual, inconsistent
- 사용하는 곳
    - 사기감지, 침입감지, 지구환경, 건강상태, 비행기안전 등
- anomaly vs classification
    - anlomaly
        - unsupervised
        - anormalouls(y=1) → very small
        - nomral(y=0) → very large
    - classification
        - superviesd
        - large number of positive and negative instances
- anamaly approach
    - descriptie task
    - predictive task
- anomaly detection method
    - model-based
    - model-free
    - global : 전체로 보면 anomaly 아닐 수 있음
    - local
- 4 types of anomaly detection approaches
    - statistical approaches
        - parametric method / non-parameteric method로 나뉨
        - parametirc method
            - gaussian 분포 → mean, std 존재
            - 모수 찾으면 됨 directly
            - 효율적
            - e(입실론)보다 작으면 이상치 판정
            - multivariate gaussian distributial
                - 다차원 분포 존재함

        - 평균벡터와 공분산 사용
        - 마찬가지로 e보다 작으면 이상치 판정
        - 대용량 데이터 행렬 역수 계산시간 엄청남 → 비효율적일수도
      - normal 안따르면 → 전처리로 normal로 바꿔줘야함
    - non-parametric method
      - build a histogram
      - 새로운 instance를 bin에 넣어서 이상치인지 판정함
      - 칸 두께 너무 좁으면 이상치로 판정될 확률 높음
      - 칸 넓으면 이상치로 판정되지 않음
    - 장점
      - 이론적 근거 good
      - 표준 통계학 기법 사용가능
      - 효과적
    - 단점
      - 문제 있는 모델이 선택될수도
      - 다차원에 대한 분포 많지 않음
  - proximity-based approaches
    - model-free
    - distance-based outlier detection
      - x is normal
        - low value of dist(x,k)
      - x is anomalus
        - high value of dist(x,k)
      - 대체안 → 덜 치우친다
        - average distance
        - median distance
    - density-based outlier detection
      - n : number of instances
      - v(d) : the volume of the neighborhood
      - d is too small
        - 죄다 anomaly로 판정
      - d is too large
        - anomalus도 noraml로 판정됨
      - x is normal

        - high value of density(x,k)
    - x is anomalus
        - low value of density(x,k)
- density-based와 distance-based 서로 역수관계에 있다
    - k번째 멀리 있다 → 주변 밀도 낮다
    - k번째 가까움 → 주변 밀도 높다
- relative density-based anomaly score
    - 주변애들이랑 같이보면 anomaly아닐수도 anomaly일수도 있음
    - relative density = 내주변 k개 애들의 평균 density/k / 내주변 density
    - relative density가 높으면 anomalous
- 장점
    - 어떤 분포이든 상관없음 → 거리, 밀도 기반이기 때문에
    - proximity measure 잘 정해야함
    - 직관적이고 이해하기 쉬움
- 약점
    - 계산 비용 큼
    - 거리 measure 어떤 거 선택?
    - parameter(d, k) 정하기 어려움

- clustering-based approaches
    - anomaly type
        - not fit the clustering well
        - small cluster
        - small in size
        - distant from normal clusters(중앙점과 멀수록 anomaly)
    - method
        - absolute distance
        - relative distance
            - 각각 거리 다구함
            - average or mean 구함
            - 나와의 거리 비교
            - 나와 centroid 거리 / 평균및 중앙값 거리 → 높을수록 anomaly score 높음
    - outlier에 민감함
        - 해결) k-means --

        - cnetorid 밀어넣고 → 재조정 → 거리계산 → 너무 멀면 제거
    - potential outlier로 취급해서 다시 밀어넣고 재조정가능
- 문제
    - the number of cluster 몇개로 할것인가?
        - 전략1) different number of cluster로 여러번 반복
        - large number of small cluster → k 크게 잡고, cluster 작게
            - cohesive
            - 작은 소규모 outlier도 정상 취급 받을 수도 있음
- 장점
    - unsupervised setting
    - normal 따로 모아서 학습할 필요 없음
- 단점
    - 클러스터 개수 몇개?
    - outlier 의 존재여부

- reconstruction-based approaches
    - 차원 낮춰서 패턴 찾아냄
        - 고차원 → 저차원
        - 저차원 → 고차원 —> reconstruction error 발생
    - k < p-demensional
        - p차원 데이터를 → pca 사용해 k차원으로 압축
        - 다시 원래 p차원으로 돌려 놓음 → reconstruction error 발생
        - reconstruction error
            - 작으면 원래 데이터와 비슷
            - 크면 원래 데이터와 멂
        - pca 한계
            - nonlinear 패턴 찾지 못함
        - autoencoder
            - multi-layer neural network
                - hidden representations
                - unsupervised setting
            - encoding(차원줄이고), decoding(차원늘림)
                - decoding 단계에서 차이가 크면 anomaly
            - backpropagation 알고리즘 사용하면 됨
                - 원본데이터가 잘 복구되어야함

- 장점
  - normal데이터만 가지고 모델링 가능 → unsupervised
  - 분포 따질 필요 없음
  - 차원 감소 기법
  - 관련 없는 속성은 알아서 무시해줌
  - autoencodef 사용해 comlex and nonlinear 패턴 찾아낼 수 있음