빅데이터 입문 중간고사 정리

1. MapReduce
   1. OverView
      * 빅데이터의 3요소 버라이어티 벨로시티 볼륨 중 볼륨에 초점
      * 클러스터 = 여러 컴퓨터를 연결 시켜주는 방법

클러스터를 구현하기 위해서는 switch 필요함

위쪽 Switch는 밑의 데이터 연결량보다 더 커야함

* + - Map = 여러 리스트 단위로 동일 연산 수행 ( 병렬수행 )
    - Reduce = 리스트를 연산으로 합치는 과정 ( 모든 맵이 끝나야 수행 )
  1. Model
     + Word Count = key 값으로 단어, value값으로 단어의 카운트
     + 이때 텍스트 파일을 여러 개로 나눠 동시에 맵 연산
     + Exection time = 모든 맵이 끝날때까지 Reduce가 기다리는 시간
     + 진행과정 : Map ( 각 리스트단위로 단어 카운트 ) -> Group by Key ( 모든 리스트끼리 Key값으로 정렬 ) -> Reduce ( 같은 키에 대하여 합침 )
     + Host Size ( html url 의 크기 )
  2. 프레임 워크의 업무
     + 데이터 나누기 ( 맵 이전 수행 )
     + 자동 스케쥴링
     + Group by Key
     + 실패 대처
     + Reduce 로 결과 전달시 랜선 뽑히면 망하니깐 전달을 확인하는 소통도 관리
  3. Master
     + Map, Reduce, worker들을 관리
     + Map이 고장나면 recovery
     + Reduce는 순차적인 업무라 중간 오류나면 처음부터 다시 수행
     + Master 가 고장나면 개 ㅄ
  4. Backup Tasks
     + 예비 인력은 냅둬서 완료에 가까운 시점에서 스냅샷을 찍어 예비인력들에게 모두 그 업무 하라고 줌 그 후 빨리 끝난애 선택받음
     + 하지만 리소스 존나 많이 먹음
  5. Combiners
     + 모든 Map task를 단순 셔플 후 Reduce에게 전달 하지 말고 저금 압축해서 보자면 communication 도 감소하고 reducer의 업무도 감소함
     + 약간 reducer와 비슷하게 보이지만 오직 하나의 mapper에게만 동작함

1. Stream
   1. OverView
      * 빅데이터의 3요소 버라이어티 벨로시티 볼륨 중 벨로시티에 초점
      * 모든 데이터는 실시간으로 끊임 없이 들어옴
      * 모든 데이터의 경향은 파악할 수 없다. ( 예측 불가능 )
   2. Model
      * 실시간 검색어 마이닝
      * 실시간 조회수 마이닝
      * SNS 피드 마이닝
      * IOT
   3. 스트림 데이터에 대한 연산들
      * 데이터 샘플링 : sliding , sampling
      * Sliding windows의 쿼리들
      * 데이터 스트림 필터
      * 서로다른 요소 카운트
      * 추산
   4. Smapling
      * 고정 퍼센트 = 퍼센트 고정
        1. 샘플링의 기준 = 데이터 튜플의 key
        2. 해쉬 함수를 통해서 해쉬값이 특정 값인 것만 추출
        3. 메모리 낭비, 초과 방지 불가능
      * 고정 개수 = 개수 고정
        1. 실시간으로 들어오는 튜플에 대한 선정 퍼센트 존재
        2. 지금 선택된거가 나갈 퍼센트 존재
        3. 먼저 다 채우고 그담꺼부터는 s/n의 확률로 pick pick되면 랜덤 replace
        4. 이때 n = time, s = memory size
   5. Sliding Window
      * 데이터 보는 창 = window
      * 분석할 데이터 크기 ( window크기 ) > 메모리 이때는 추산
      * 추산 방법
        1. 지수 window
           1. 지수 단위로 증가하는 window
           2. 같은 크기의 비트는 항상 2개 다음 비트는 이전 비트 크기 윈도우 2번째 시작점부터 출발
           3. 언제나 끝지점은 마지막 요소의 2번쨰 구간중 하나에 있어야함
        2. DGIM
           1. 원하는 튜플 개수에 따라 윈도우 사이즈 결정
           2. 즉 1카운트 문제이면 처음 1부터 원하는 개수가 나오는 구간까지를 윈도우 설정
           3. Bucket = 시작점과 우너하는 튜플의 개수 저장 공간 ( 끝점 X )
           4. 큰거는 옛날 데이터
           5. 항상 원하는게 들어오면 윈도우 1짜리 만들고 만약 같은 크기 bucket 3 개 된다면 뒤의 bucket merge 계속 반복
        3. 오류는 1/r 하지만 r을 키우면 메모리 증가
   6. Filtering
      * 어떤 키 기준으로 필터링 하고 싶을 떄 사용
      * 단순 해쉬 테이블을 이용해 하기에는 만약 10억개의 메일중 스팸 처리하기 위해서는 10억개에 대한 해쉬값을 저장할 해쉬테이블 있어야하니 절대 못써
      * Bloom Filter
        1. 여러 해쉬 함수의 해쉬값에 대하여 결과를 비트 표현
        2. Setup Process 각각의 해쉬값을 비트 표현 후 그 값들 합집합 or 연산
        3. LookUp Process 확인하고 싶은거 해쉬값 비트 표현 후 모든 1이 bit array의 1자리에 속한가?? 그럼 높은 확률로 ok!
        4. 특징 false positive 는 존재하지만 false negative는 절대 불가능

이 특징이 존나 중요해 그니깐 아닌걸 맞다 할수 있는데 맞는걸 아니다라고 할수 없다.

* + - 1. 성능 = 해쉬 개수가 적당히 존재해야함 2~4
  1. Counting Distincy Elements
     + 버리자
  2. Computing Moments
     + Counting Distinct Element가 이거의 스페셜 케이스
     + AMS method
       1. 적률 = 값이 작을수롱 균등함

1. 통계
   1. Terminology (경향)
      * Population = 타겟 데이터 셋
      * Sample = 전체 데이터에서 우리가 수집한 데이터
      * Descriptive 통계 = 정보를 요약한거 ( 평균 )
      * Inferential 통계 = 결론의 신뢰성을 측정하기 위한 방법들 ( 신뢰구간 )
      * Parameter = 숫자적으로 활용될 수 있는 알려지지 않은 값 ( 알파, 베타, 변수 )

그냥 데이터 셋에서 특정 경우의 데이터들

* + - Statistics = 알려진 값 즉 파라미터를 실제 값으로 파악하기 위해 특정 계산을 한거 -> 어떤 데이터 셋에서 특정 어떤 방법을 통해서 특정 경우의 데이터들을 구하겠다. ( 즉 파라미터에 방법이 추가 )
    - Discrete = 유한한 숫자 ( 하루 업로드 트윗 수 )
    - Continuous = 무한한 숫자 ( 시간 )
  1. Central Tendency ( 중점 경향 )
     + 평균
     + 중간값 : 정렬 후 중간 인덱스 값
     + 모드 : 가장 많이 나온 값
     + Variation = 어느정도 분포가 되어있나?
       1. Range : 최대 – 최소
       2. Percentile : 전체 리스트를 구간으로 나눈거 ( 상위 10퍼센트 )
       3. Quartiles : 4등분한거 ( 상위 25퍼센트 )
          1. Q1 = 상위 1 2 구간의 경계
          2. Q2 = 상위 2 3 구간의 경계
          3. Q3 = 상위 3 4 구간의 경계
          4. Interquartile Range = Q1 Q2 Q3 min max 사용해서 Q1 ~ Q3 사이의 구간
          5. Five-num-sumary = Q1, Q2,Q3 , min, max
       4. Variance = 분산
          1. 좀더 강화된 평균
  2. Probabilty = 추산
     + PMF = probability mass ft
       1. 구분된 랜덤 값의 동작을 x , 각 동작의 확룔 = y
       2. 즉 Discerte 한 데이터를 다룸
       3. 모든 확률 더하면 1
       4. P(x=xi) = 1/k 즉 k개의 x가 존재
     + PDF = continuous random variable
       1. 이번에는 구간의 확률을 파악 즉 continuos한 데이터를 다룸
       2. 똑같이 모든 확률 더하면 1
       3. A~b 구간의 면적은 1
     + Sum Rule = 한 확률은 그 확률이 일어나고 다른 사건이 일어날 확률 다 합친거
     + Conditional Probabillity = 조건부 확률
       1. P(y|x) = P(y,x)/P(x) ( x가 일어날 때 y가 일어날 확률 = 모두 일어날 확률 / x일어날 확률 )
     + Bayes’ Rule
       1. P(x|y) = P(x) )/P(y) \* P(y|x)
     + Chain Rule = 조건부 확률 P(x,y) = P(y|x)P(x) 이거 사용한거 ㅇㅇ
     + Independence = 독립 확률
     + Expectation = 기대값
       1. Ex~P[f(x)] = x가 P일 때 f(x)의 기대값 -> 즉 P(x)f(x) -> x의 확률 \* 값 다 더한거
       2. 선형계산 가능
     + Covariance = 공분산 = 얼마나 같이 변했나?
       1. (실제값 – 기대값)(실제값 -기대값)
       2. 중요한거는 음수일수도 있어
     + Randomize Algorithms = Monte Carlo 법칙
       1. 랜덤하게 뽑으면 얼추 전체랑 비슷함 ㅇㅇ

1. Distribution = 분포
   1. Brenoulli Distribution 베르누이
      * 동전 뒤집기 확률
   2. Gaussian Distribution 가우스
      * 종모양 분포
      * 분포가 평균 기준으로 좌우가 쓱쓱 같다
   3. Multivariate Gaussian 여러 변수 가우스
      * 3차원 가우스 ㅇㅇ 메트릭스 ㅇㅇ
      * 여러 개가 합쳐서 하나의 가우스 분포
   4. Gaussian Mixture
      * 여러 개의 가우스 분포가 나옴
   5. Sigmoid 함수
      * 어떤 데이터든 0~1 사이로 알려줌 ㅇㅇ
   6. Softplus 함수
      * 0보다 작으면 0 크면 무한히 커짐
2. Information Theory = 정보이론 ( 정보가 얼마나 많이 표현되어있나? )
   1. 일반적으로 발생하지 않는 경우를 학습 하는게 일반적으로 발생하는 경우를 학습 하는 것보다 훨씬 유용한다.
   2. 정보이론은 즉 일반적으로 발생하지 않는 경우가 어느정도 있나?
   3. -log(P(x)) 즉 확률이 작을수록 값이 커짐 ㅇㅇ
   4. 엔트로피 = 불확실성
      * 얼마나 불규칙한가? 얼마나 정보가 없나?
      * 값이 높으면 불규칙적이다. -> 표현하기 어렵다 -> 표현하기에 데이터 비트가 많이 필요하다
   5. KL Divergence
      * 두개의 변수가 얼마나 차이가 있나?
      * 항상 음수임
      * 같으면 0
      * 교환법칙 적용 안된다.
   6. Cross Entropy
      * H(P,Q) = 엔트로피 + Divergence <- 두개의 변수가 얼마나 차이가 있나?
      * 즉 교환법칙 적용이 된다.
      * KL Divergence의 진화버전
3. Association btw Dist = 두개의 데이터셋간의 관련성
   1. Pearson Correlation Coeff = 피어스 상관관계 계산
      * 값이 1 이면 모양이 같다.
      * 값이 0 이면 연관 없다
      * 값이 음수이면 하나가 증가하면 다른거는 감소다
      * 단점 = 평균에 기반하니깐 정규분포를 따라야 사용 가능함
   2. Kendall’s Tau = 랭크에 대한 상관관계 계산
      * 각 데이터의 랭크간 상관관계 계산
      * 1 = 모양이 같다
      * 0 = 연관 없다
      * 음수 = 반대
      * 단점 : 서로 다른기준으로 랭킹 한거 두게 데이터들 간 어느정도 교집합인 데이터가 있어야함.
      * 동일한 사이즈 동일한 데이터가 순서만 다를때 캔다스 타우를 사용할수 있다만 기억
4. Hyphothesis Testing = 통계의 꽃 ( 가설 검증 )
   1. 가설 검증 방법
   2. Null hyphothesis = 우리가 원하는 가설의 반대
      * P-value = 커지면 귀무가설을 잡게됨 가설이 틀렸다라는거를 증명
        1. 데이터가 작으면 항상 높으니 신뢰성 낮아
      * T-test = 평균으로 p벨류 구하는거
        1. 첫번쨰 값은 차이를 나타냄 , 두번째값은 p-value 값이 크면 가설 틀림!
   3. Alternative hyphothsis = 우리가 원하는 가설
      * ㅋ
   4. Mann-Whitney U test = 평균을 사용 안해서 사용하니깐 데이터가 작아도 ㄱㅊ
   5. Type Error = Null 가설결정하는데 틀린거 즉 기각해야할 가설을 잘못 선택한거
      * Error1 = 기무가설이 맞는데 기각한거
      * Error2 = 기무가설이 틀렸는데 승인한거
   6. Anova Testing = 즉 검증 여러 개간 ㄱㅊ은지??
   7. K-S ㅅtwo side = 아노바 양방향으로 한거 ㅇㅇㅇㅇㅇㅇㅇㅇ