데이터 마이닝을 위한 대용량 고차원 데이터의
셀-기반 분류방법

전북대학교 컴퓨터 과학과
(dsjin, jwchang)@dblab.chonbuk.ac.kr

Cell-based Classification of High-dimensional Large data for Data
Mining Application

Du-Seok Jin, Jae-Woo Chang
Dept. of Computer Engineering, Chonbuk National University

1. 서론

데이터 마이닝은 대용량의 데이터베이스에서 습득한
지식, 패턴, 연관된 규칙 등을 발견하는 방법이다. 데이터
를 분석하여 서로 유사한 그룹으로 데이터를 분류
하는 방법은 데이터 마이닝의 중요한 분야 중 하나이다.

입력 데이터, 트레이닝 셋(Training Set)은 다수의 에러
데이터를 가진 대규모의 데이터 셋이다. 분류는 입력 데이터
를 분석하여 데이터에 나타나는 특성에 맞는 모델을
설정하는 일이다. 분류(Classification)모델을 설정하기 위
하여 첫째, 입력 데이터의 클러스팅(Clustering) 과정이
필요하다. 클러스팅 기법은 클러스터들 사이에 계층구조
를 가지고 있는 계층적(hierarchical) 클러스팅법과 데이터
공간을 정의할 수 있는 클러스터링에 따라 분류하고 클러스
트링하는 공간분할(space partitioning) 클러스팅방법
이 있다. 그리고 클러스팅을 결정하는 기준은 거리
distance)를 사용하는 방식과 밀도(density)를 사용하는
방식으로 구분된다. 기존의 클러스팅 알고리즘은 고차원
데이터의 클러스팅영상에서는 적합하지만 차원이 높아지수록
규격된 성능 저하를 보인다. 그러나 첫째, 고차원 형태의
data가 증가하고 있기 때문에 고차원 데이터를 효율
적으로 처리할 수 있는 분류방법이 요구되고 있다. 각
분류 모델을 설명하기 위한 기존의 분류 알고리즘은 단지
메모리 상주(memory-resident) 데이터에 대해 적용되어
있어 대용량 데이터를 효율적으로 처리하지 못하고 있다. 분류모델은 Decision Tree[1], 통계모델[2], 유전학적 모델[3]이 지금까지 연구되고 있다. Decision Tree는 입력
데이터를 하나의 그룹에 속할 때까지 재귀적으로 분할
하려 트리를 구성한다. 이때 효율적인 분할 인덱스
(Splitting Index)에 대한 연구가 분류기법의 성능을 결정하
는 중요한 역할을 한다. 따라서 본 논문에서는 gini
index[4]에 기반한 셀-기반 분류모델에 적합한 분할
인덱스를 정의하여 대용량 데이터를 처리하기에 적합한
분류 알고리즘을 제안한다. 또한 데이터의 숫자값이 수
치(numerical)크리스트리트 또는 범주(categorical)크리스트리
트에 따라 기존의 연구에서는 추가적인 작업이 필요하지
만 제한하는 방법에서는 별도의 처리과정이 필요하지
않다. 아울러, 본 논문에서는 데이터 공간을 각 차원별 구
간으로 분할하고 각 분할구간의 밀도를 구하여 저장하는
방식으로 고차원 데이터를 효율적으로 처리할 수 있는 방
법을 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 논문
에서 제안하는 셀-기반 분류방법의 공간분할 및 분류결
과를 설명하고 심층화하는 방법에 대해 설명하고 제 3장
에서에는 셀-기반 분류방법의 성능을 제시한다. 마지막으로
4장에서는 결론 및 향후 연구과제를 제시한다.

2. 대용량 고차원의 데이터를 위한 셀-기반 분류방법

본 장에서는 대용량 고차원의 데이터를 효율적으로 클러스
트링하고 분류하기 위한 셀-기반 분류방법을 제안한다.
이들 통해 공간상에서 분할 인덱스(splitting index)를
통해 각 차원의 구간을 정하고 각 구간에 따라 셀을 구성
한다. 얻어진 셀의 밀도가 높은 것 혹은 이하인 셀을 저장하
고 검색하는 방법을 제안한다.

2.1 공간분할 및 분류된 결과 저장
공간상에 표현된 입력데이터를 해당되는 셀로 정렬하기 위해서 먼저 각 차원에 대해서 N 개의 구간으로 나눈다. 이때, 차원 셀을 계속해서 2N 개로 구간을 나눈지 를 분할 인덱스를 통해 결정한다. 트레이닝 데이터에 대해서 차원을 N 개로 분할할 경우 각각의 분할영역에서 C 개의 클래스에 대하여 상대적인 밀도의 1/n을 구하 여 모두 합한 결과를 가지고 결정한다. 분할 인덱스는 식 1로 구한다. 그림 1은 입력 데타가 2차원인 경우의 분할과정을 설명하며, 그림 1에서 실선으로 나타난 부분이 분할구간이 된다.

\[ \text{Splitting Index} = 1 - \sum_{i=1}^{N} \sum_{j=1}^{n} p_{ij}^2 \quad (\text{식 1}) \]

그림 1. 분할인덱스를 통한 공간분할

트레이닝 샘플을 통해 분류된 결과를 저장하는 방법은 다음과 같다. 첫째, 모든 샘플의 빈도수를 구하여 분류정보 파일(Classification Information file)에 분류 결과 및 빈도수를 저장한다. 이 방법은 샘플의 빈도수를 분류정보 파일의 오프셋으로 사용하여 질의 처리 시 반복의 I/O로 결과와 비교할 수 있으나 샘플의 수가 많은 경우 분류정보 파일의 크기가 매우 큰 단점이 있다. 둘째, 모든 차원에 대한 각각의 구간별 빈도수를 구하여 구간 임계값 이상인 구간과 미만인 구간을 구분하여 근사정보파일(Approximation Information file)에 0에서 저장하고, 분류정보파일에는 각 샘플에 대해 빈도수에 빈도수의 셀 임계값 이상인 셀의 빈도수를 저장한다. 셀 임계값과 구간 임계값은 식 2와 같다.

\[ \lambda = \frac{NR}{NI} \times F \]

간임계값(\(\lambda\)) = (식 2)

임의분할의수(\(NI\)) = 입력데이터의수
비교차원의분할구간(\(F\)) = 각차원별각구간의최소빈도수

3. 실험 및 성능평가
본 장에서는 제안된 셀-기반 분류방법을 구현하여 성 능 평가를 수행한다. 실험을 통하여 제안된 셀-기반 분류 방법이 대용량 데이터에 대한 처리가 가능함을 보이고, 각 차원의 구간별 밀도에 대한 구간 임계값을 변화하면 서 성능을 측정한다. 실험에 사용된 시스템 환경은 CPU
650 MHz dual, 메모리 512MB 의 리눅스 서버에서 수행하였다. 사용된 데이터는 IBM Quest Data mining project[5]에서 사용한 Synthetic Data Generation Code for Classification을 이용하여 각각 4차원, 8차원의 10만 건의 데이터를 사용하였다. 성능평가는 10 만 건의 데이터를 분류하여 근사정보파일과 복잡정보파일을 생성하는 시간과 1000 건의 질의레코드를 처리할 때 구간 임계값에 따라서 분류정보파일 검색하지 않고 근사정보파일에서 설계는 비율과 정확음을 측정한다. 분류시간은 4차원의 경우 약 6초 정도 소요되며 8차원의 경우 약 300초 정도 소요된다.

![그림 3: 필터링 비율](image3)

그림 3은 각 구간 임계값에 따라 근사정보파일에서 여파되는 비율을 나타낸다. 구간 임계값을 크게 하면 여파되는 비율은 점점 커진다. 그러나 정확성이 떨어지는 단점이 있다. 반대로 구간 임계값을 작게 하면 정확성이 높으나 여파되는 비율이 적어 평균검색 시간이 느려지는 단점이 있다.

![그림 4: 정확율](image4)

그림 4는 구간 임계값에 따라 1000 건의 질의에 대한 정확율을 나타내며, 그림 5는 구간 임계값에 따라 1000 건의 질의에 대한 평균 검색시간을 나타낸다. 구간 임계값이 커질수록 검색 시간이 빨라 정확율이 떨어진다. 따라서, 정확율과 평균 검색시간의 상호관계를 고려한 본 논문에서 제안한 방법의 성능은 식 3으로 표현할 수 있다.

\[ E(L) = P(L) \times Wp + T(L) \times Wt \]  (식 3)

식 3에서 Wp는 정확율의 가중치이며, Wt는 검색시간의 가중치이다. P(L)는 구간 임계값이 L일때의 정확율을 나타내며, T(L)는 구간 임계값이 L일때의 평균검색시간을 나타낸다. P(L)와 T(L)는 0~1사이의 값으로 정규화한 값이다. 그림 6은 구간 임계값에 따른 성능 E(L)를 나타낸다. Wp와 Wt 각각 1의 가중치를 주었을 때 \( \lambda = 0.2 \)일때 가장 효율적인 성능을 보인다.

![그림 5: 평균 검색시간](image5)

![그림 6: 가중치를 적용한 성능비교](image6)

4. 결론 및 향후연구
본 논문에서 제안한 설계방법은 첫째, 메모리 상주 데이터에 대해 확장되어 있는 문제점을 해결하고, 고차원의 데이터를 효율적으로 처리할 수 없는 문제점을 각 차원별 구간 임계값을 사용하여 크기가 큰 분류정보파일을 검색하는 횟수를 줄임으로써 성능을 향상시켰다. 또한, 데이터의 수준에 관계없이 적용 가능한 새로운 분류 알고리즘을 제안하였다.
향후연구로는 보다 다양한 데이터를 가지고 실제 응용에 적용하여 본 논문에서 제안한 방법의 성능을 측정하는 연구가 필요하다.

참고 문헌