



데이터 마이닝

걷히는 안개를 바라보면서

데이터 마이닝은 여러 가지로 정의할 수 있지만 쉽게 설명하면 많은 양의 데이터에 함축적으로 들어 있는 지식이나 패턴을 찾아내는 기술이다. 데이터 마이닝은 비교적 최근에 연구가 시작되고 관련 소프트웨어가 개발되고 있는 최첨단의 전산학 분야 중 하나다. 1983년에 IBM Almaden

연구소에서 Rakesh Agrawal 박사를 중심으로 Quest 데이터 마이닝 프로젝트가 시작된 이후로 선진국의 우수 연구소와 대학원을 중심으로 활발하게 연구가 되어왔다. 1994년 필자가 IBM Almaden 연구소에서 Rakesh Agrawal 박사의 지도 아래 데이터 마이닝 연구를 시작할 때만 하더라도 이 새로운 분야가 정말 성공할 수 있을지, 또 사람들을 위해 정말로 유용하게 쓰일 수 있는지 확실하지 않았다. 정말로 안개가 가득한 산속에서 어디로 가야할지 모르고 헤매는 듯한 기분으로 데이터 마이닝 연구를 시작했다. 하지만 그 뒤로 IBM Almaden 연구소와 벨 연구소에서 데이터 마이닝과 관련된 여러 가지 기술을 개발했고 논문을 썼으며 또 미국 특허들을 취득하거나 신청했다. 그러는 가운데 데이터 마이닝에 관한 필자의 안목도 조금씩 넓어졌다. 뿌연 안개 속에서 헤매는 것 같았던 처음에 비하면 이제는 필자에게는 그 안개가 하나

특집 1부

인쇄술의 보급이 지식의 양과 속도를 그 이전 시대에 비해 엄청나게 증가시켰듯이 최근 20여년의 컴퓨팅 환경의 발전은 인간이 따라잡기 힘들 정도로 엄청난 양의 지식을 빠르게 쏟아내고 있다. 이제는 단순한 사실이나 자료가 아닌 인간이 이해할 수 있는 형태의 정보와 지식이 더 중요한 시대가 된 것이다. 그리고 그 변화의 한가운데 데이터 마이닝이 있다고 해도 과언이 아니다.

심규석 KAIST 전산학과 교수
shim@cs.kaist.ac.kr
<http://cs.kaist.ac.kr/~shim>

씩 걷히는 것 같은 기분이다. 이제 데이터 마이닝에 관해 좀 더 확실하게 볼 수 있게 되었으므로 독자들에게 이에 대해 소개하고 이해를 돕고자 한다.

데이터 마이닝, 왜 알아야 하나

이제는 많은 회사들이 자신의 비즈니스에 관련된 여러 가지 데이터를 모아 데이터베이스 시스템에 넣어두고 있고 이 데이터의 양은 해마다 끊임없이 증가하고 있다. 또한 인터넷과 전자상거래가 급속하게 보급되면서 소비자 구매에 관련된 많은 양의 데이터가 자동으로 컴퓨터에 모이게 됐다. 이로 인해 과거에는 가능하지 않았던 거대한 양의 데이터를 우리 주변에서 쉽게 찾아볼 수 있는 시대가 됐다. 하지만 이렇게 모아놓은 데이터로부터 아주 유용한 정보를 찾아내 마케팅이나 회사의 이익을 효율적으로 증대하기 위해 사용하는 데는 아직도 어려움이 많다. 그 이유 중 하나는 이 정보가 아주 많은 양의 데이터 안에 함축적으로 숨어 있어 사람의 눈으로 일일이 조사하는 것이 불가능하기 때문이다. 다행히도 데이터 마이닝 분야에서 개발된 기술을 통해 이러한 데이터로부터 유용하고 값진 정보를 효과적으로 찾아내 회사뿐만 아니라 개인의 일상생활도 편리하게 도와줄 수 있게 됐다.

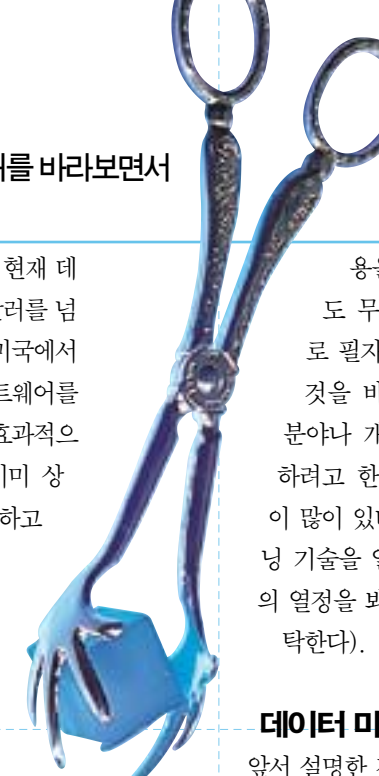
가트너 그룹(Gartner Group)은 데이터 마이닝 기술을 이용한 타겟 마케팅(target marketing)이 아직까지 일반 회사에서 5%도 안되게 쓰이고 있지만 10년 내로 80%

이상 사용될 것이라 내다보고 있다. 현재 데이터 마이닝 시장의 크기도 10억 달러를 넘었다고 보고 있다. 그렇기 때문에 미국에서는 많은 상용 데이터 마이닝 소프트웨어를 개발해 팔고 있고, 여러 분야에서 효과적으로 이용되기 시작했다. 국내에도 이미 상륙해 국내 시장을 향해 힘차게 돌진하고 있다.

하지만 국내에서는 아직 데이터 마이닝에 대한 이해나 노력이 별로 이뤄지지 않고 있는 실정이다. 그래서 데이터 마이닝 기술이 어떤 유용한 일을 우리에게 해줄 수 있는지도 사람들이 잘 알지 못할 뿐더러 어떤 소프트웨어를 써야 할지 모르는 경우도 많다. 이런 시점에서 부족하나마 데이터 마이닝의 본고장에서 실제 소프트웨어를 개발했던 필자의 경험을 바탕으로 기본적인 개념과 응용 분야에 관해 간단하게 설명하고, 어떤 상용 소프트웨어가 있는지 중요한 판매사(vendor)와 실제 사용된 예를 소개하고 독자들에게 자신의 일터에서 어떻게 사용할 수 있는지에 대한 기본적인 아이디어를 제공하려고 한다. 데이터 마이닝 기술 자체가 너무나 방대해 모든 것을 나열하기에는 어려운 점이 많다. 또한 데이터 마이닝 기술을 자기 회사의 이익을 위해 아주 효과적으로 사용하고 있는 회사라 하더라도 경쟁사에 정보를 제공하지 않기 위해 사용 예와 얼마나 많은 효과를 얻고 있는지에 관해 제대로 발표하지 않는 경우가 많아 그

걷히는 안개를 바라보면서

특집 1부



용을 자세히 알아내기도 무척 힘들다. 그러므로 필자는 그 동안 경험한 것을 바탕으로 알고 있는 분야나 개발된 기술을 서술하려고 한다(혹시 부족한 점이 많이 있더라도 데이터 마이닝 기술을 알리고자 하는 필자의 열정을 봐서 많은 이해를 부탁한다).

데이터 마이닝이란

앞서 설명한 것처럼 데이터 마이닝은 많은 양의 데이터에 함축적으로 들어 있는 지식이나 패턴을 찾아내는 기술이라고 정의할 수 있다. 데이터 마이닝 기술은 백화점에서 물건을 진열할 때 고객의 움직임을 줄여주기 위해 활용할 수 있고, 고객의 구매 패턴을 보고 유용한 패턴을 찾아내 소비자가 살 물건을 미리 예측하고, 쿠폰을 발행해 관심을 유발함으로써 판매를 촉진할 수도 있다. 보험 회사에서는 고객이 다른 회사로 옮기는 것을 방지하거나 고객의 위험성에 따라 보험료를 차등화해 제공하는 데 사용할 수 있다. 또 신용카드 회사에서는 훔친 신용카드를 사용하는 경우를 발견해 더 이상의 불법 사용을 막거나 새로운 고객이 신용카드를 신청할 경우에 카드 발급 결정에 사용할 수도 있다.

전자상거래를 위한 웹 서버인 경우에는 소비자가 방문한 웹 페이지와 구매한 물건과 소비자의 특징을 보관하고 있기 때문에 이 데이터를 분석하면 각각의 사용자에게 맞는 웹 페이지를 동적으로 그때 그때 생성해 주거나, 웹 페이지의 캐싱(Caching), 프리페칭(Prefetching), 스와핑(Swapping)을 효율적으로 제공할 수 있어 성능을 높이고 수행속도를 빠르게 할 수 있다. 더욱이 웹 액세스의 다차원 웹 로그 분석을 이용한 트랜드 분석을 통해 웹에서 어떤 일이 일어나고 있는지에 대해서도 대략적인 정보를 제공해줄 수 있다. 또한 모든 소비자에게 동일한 웹 페이지를 제공하는 것이 아니라 소비자의 관심에 따라 다른 웹 페이지를 동적으로 만들어 제공하는 개인화(personalization)

데이터 마이닝의 대부, Rakesh Agrawal 박사

1983년도에 University of Wisconsin at Madison에서 박사학위를 받고 벨연구소에서 1983년부터 1989년까지 Ode라는 객체지향형 데이터베이스에 관해 연구했다. 그 후 IBM Almaden 연구소에서 일했고 1993년도부터 Quest 데이터 마이닝 프로젝트를 시작하고 주도했다. 그는 데이터 마이닝이라는 분야가 개척되고 자리잡는 데 크게 기여한 인물이다. 그가 개발한 데이터 마이닝 기술로는 OLAP, 연관규칙, 순차 패턴, 분류, 군집화, 유사시 계열 시퀀스(Similar Time Sequences), 텍스트 마이닝, 데이터베이스 마이닝 통합(Database-Mining Integration) 등이 있고, IBM Intelligent Miner라는 데이터 마이닝 소프트웨어의 개발에 주도적인 역할을 하기도 했다. 그는 데이터 마이닝 분야에서 학문적인 것 뿐만 아니라 상용 소프트웨어의 개발에 기여한 공로를 인정받아 2000년도에는 ACM SIGKDD Innovation Award를 수상했다(<http://www.almaden.ibm.com/cs/people/ragrawal/bio.html> 참조).

특집 1부

서비스를 가능하게 할 수도 있다.

네트워크 분야에서는 네트워크에 이상이 생기기 전에 과거의 네트워크에 관련된 데이터를 이용해 앞으로 몇 시간 안에 네트워크에 생길지도 모르는 문제를 미리 예측해 낼 수도 있다. 피자헛 가게를 새로운 장소에 개점할 경우에 과거의 다른 피자헛 가게가 세워진 곳에 관련된 정보로부터 새로 세우는 장소에서 성공할지를 예측하는 데도 사용할 수 있다.

교차 판매(cross-selling)나 상승 판매(up-selling) 등을 통해 회사의 판매 실적을 더 높일 수도 있다. 교차 판매란 서로 다른 부류에 속하는 상품이지만 서로 연관돼 고객들이 구매하는 경우를 찾아 연관된 상품을 고객에게 추천해 판매하는 것을 뜻한다. 예를 들어 장난감을 사는 고객이 생명보험에 들 가능성이 많다면 장난감을 사는 고객에게 생명보험에 관한 정보 제공해 보험에 가입할 수 있도록 만드는 것을 말한다. 상승 판매란 1억원의 생명보험을 가입하려는 고객에 대한 정보를 분석해보고 만일 그 고객이 2억원짜리 보험에 가입할 가능성이 많은 고객이라면 2억원의 보험에 대해 같이 소개하고 추천해 더 비싼 보험을 들도록 유도하는 것을 말한다. 이 밖에도 여러 분야에서 데이터 마이닝 기술을 유용하게 사용할 수 있다.

대표적인 데이터 마이닝 기술

현재 상용 데이터 마이닝 소프트웨어에서 제공되는 주요한 알고리즘 중에서 필자가 생각하기에 최근에 개발된 중요한 기술을

〈표 1〉 구매 물품 자료

구매 번호	구매 항목
1	{라면, 오렌지 주스, 커피}
2	{라면, 소시지}
3	{라면, 커피}
4	{오렌지 주스, 비누, 삼푸}

〈표 2〉 고객 신상 명세 데이터

고객 번호	나이	결혼 여부	자가용 수
1	23	No	1
2	25	Yes	1
3	29	No	0
4	34	Yes	2
5	28	Yes	2

소개하자면 연관 규칙, 순차 패턴, 분류, 군집화, 아웃라이어 판별 등이 있다. 이 중에서 데이터 마이닝 알고리즘의 대표적인 예가 될 수 있는 연관 규칙과 관련된 알고리즘부터 설명하고 나머지 다른 기술들에 관해 간략하게 소개하겠다.

연관 규칙(Association Rules)

데이터 마이닝을 소개할 때 대표적으로 언급되는 기술로 백화점이나 슈퍼마켓에서 한번에 함께 산 물건들에 관한 연관 규칙을 찾아내는 기술이다. 실제 데이터를 이용해 발견했던 아주 유명한 연관 규칙 중 하나는, 미국의 대형 편의점의 소비자 구매 데이터에 이 기술을 적용한 결과, 아기 일회용 기저귀를 사는 사람은 맥주도 같이 산다는 연관 규칙을 발견한 것이다. 이러한 패턴을 발견하고 소비자들에 관해 조사해 본 결과, 보통 아기 엄마가 남편에게 기저귀를 사오라고 하면 남편이 편의점에 들러 기저귀를 사면서 같이 맥주도 사간다는 것을 발견했다. 이러한 연관 규칙이 발견됐을 경우 맥주의 판매를 늘리기 위해 일부러 기저귀 값을 할인해 더 많은 맥주가 팔리도록 할 수 있다.

이 기술을 사용하기 위해 가장 보편적인 입력 데이터의 각각의 원소는 슈퍼마켓에서 한 번에 사는 장바구니에 들어있는 상품들이라 할 수 있다. 이 때 장바구니 하나에 들어가는 데이터의 집합을 한 **트랜잭션**(transaction)이라 한다. 〈표 1〉에 네 개의 트랜잭션으로 구성된 구매 데이터를 테이블로 나타냈다. 첫 번째 소비자는 라면, 오렌지 주스, 커피라는 세 가지 물건을 샀고, 두 번째 소비자는 라면과 소시지를 함께 구매했다. 연관 규칙 기술을 적용할 때 두 가지 설정값을 입력해야 하는데 이들은 **지지도**(support)와 **신뢰도**(confidence)라 불린다. 어떤 규칙의 지지도가 10%라면 그 의미는 전체 트랜잭션 중에서 그 규칙을 따르고 있는 트랜잭션이 10%를 차지한다는 것을 의미한다. 예를 들어 〈표 1〉의 경우, {라면} → {커피} 라는 연관 규칙은 '라면을 산 사람은 커피

도 같이 산다' 는 의미인데, 네 가지 트랜잭션 중 1번과 3번 소비자가 구매한 물건들에 들어 있는 규칙이므로 지지도는 50%가 된다. 그리고 신뢰도는 규칙의 왼쪽에 있는 것을 산 사람들 중에서 오른쪽에 있는 물건들을 모두 산 사람들의 퍼센트를 말한다. 예를 들어 앞의 규칙에서 라면을 산 사람들은 세 사람인데 그 중에서 커피를 산 사람은 두 사람이므로 이 규칙의 신뢰도는 66.7%가 된다. 연관 규칙 알고리즘을 제공하는 소프트웨어를 쓸 경우에 최소 지지도와 최소 신뢰도를 데이터와 함께 설정하면 두 조건을 만족하는 모든 연관 규칙을 다 찾아낸다. 이때 연관 규칙에는 왼쪽과 오른쪽 모두 여러 개의 상품이 올 수 있다. 예를 들면 {라면, 오렌지 주스} → {커피} 라는 연관 규칙도 가능한데, 이 연관 규칙은 '라면과 오렌지를 사는 사람은 커피도 산다' 는 뜻이다.

앞에서 설명한 데이터는 불린 속성(boolean attribute)을 가진 데이터다. 다시 말하면 어떤 항목을 '구매했다' 또는 '아니다' 만을 나타낸다. 하지만 더 나아가 〈표 2〉와 같이 관계형 데이터베이스의 테이블과 같은 형식의 데이터를 이용해 연관 규칙을 찾을 수 있는데 이것을 정량적(Quantitative) 연관 규칙이라 한다. 〈표 2〉의 데이터는 고객들에 대한 구매 정보가 아니라 각각 고객의 신상 명세에 관한 데이터로 숫자 속성(numerical attribute)과 범주 속성(categorical attribute)이 있다. 여기서 결혼 유무를 나타내는 열이 범주 속성에 해당한다. 앞의 데이터에서 찾을 수 있는 연관 규칙의 예를 들면 {나이: 30...39}와 {결혼 유무: Yes} → {자가용수: 2} 인데 이 규칙은 지지도가 40%, 신뢰도가 100%다. 이 규칙은 '30대의 결혼한 사람들은 대부분 자가용을 두 대씩 갖고 있음' 을 나타낸다. 앞에서 말한 구매 데이터 외에도 이와 같이 우리 주변에서 흔히 볼 수 있



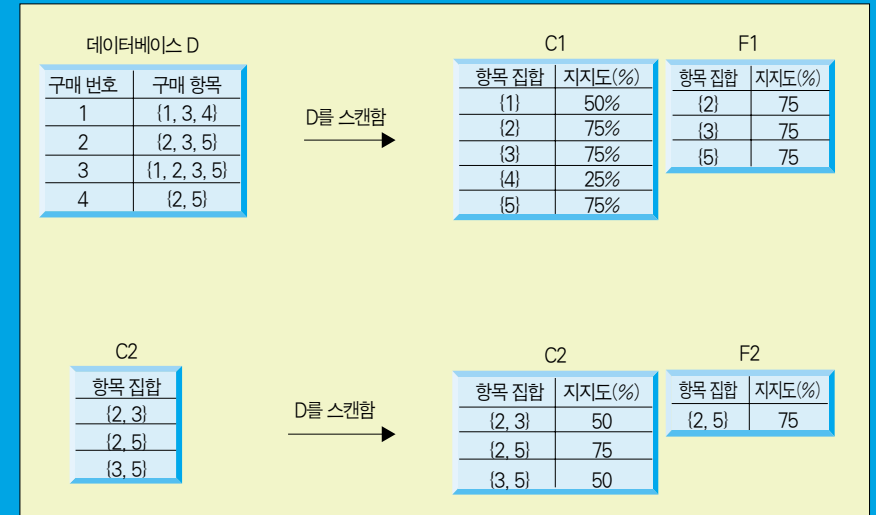
연관 규칙 알고리즘

연관 규칙을 찾아주는 알고리즘 중에서 가장 먼저 개발됐고, 또 가장 많이 쓰이는 알고리즘은 Apriori 알고리즘이다. 이 알고리즘은 두 가지 단계로 구성된다. 우선 첫 번째 단계에서는 최소 지지도 설정값에 따라 빈도수가 높은 항목의 집합들을 찾아내고 그 다음 단계에서는 이들 집합들로부터 신뢰도 설정값을 만족하는 연관 규칙을 모두 뽑아낸다. Apriori 알고리즘에서 사용하는 중요한 법칙은 빈도수가 높은 항목의 집합의 모든 부분 집합도 다 빈도수가 높다는 사실이다. 예를 들어 데이터에 {라면, 커피, 설탕}이 최소 지지도에 의해 빈도수가 높다면 당연히 {라면, 커피}만을 봐도 빈도수가 높고, {커피, 설탕}을 봐도 빈도수가 높다. 다시 말해 어떤 집합이 주어졌을 때 새로운 항목을 더해주면 지지도는 절대로 전보다 증가할 수 없다.

주어진 n개의 항목이 있을 때 이 항목을 이용해 만들 수 있는 모든 항목 집합은 2^n개가 있다. 예를 들어 {a, b, c}의 모든 부분 집합은 {}, {a}, {b}, {c}, {a, b}, {a, c}, {b, c}, {a, b, c}가 있다. 그렇기 때문에 주어진 2개의 항목에 대해 모든 부분 집합을 만든 후에, 각각에 대해 데이터를 보고 지지도 계산을 위해 카운트하려고 하면 이 부분집합의 개수가 너무 많아 모두 메인 메모리에 넣고 카운트할 수 없게 된다. 이해를 돕기 위해 설명하면, 만일 1000개의 물건을 판매하는 잡화점에서 그 물건들을 함께 살 수 있는 모든 경우를 다 따지면 2^1000이 되고 이것은 너무나 큰 숫자다. 그래서 간단하게 만드는 초보적인 알고리즘은 이 기술을 실제 생활에 사용할 수 없게 만든다.

하지만 앞에서 말한 법칙을 사용하면 그것을 가능하게 만들어 준다. Apriori 알고리즘은 우선 사이즈 한 개의 빈도수가 높은 항목들을 먼저 구하고, 그 다음에 이것들을 이용해 사이즈가 두 개인 빈도수가 높은 항목들의 집합을 구하는 방식으로 한 사이즈씩 차례로 수행한다. 그렇기 때문에 데이터에 있는 사이즈가 가장 큰 빈도 높은 항목 집합의 크기가 k라면 대략적으로 데이터를 k번 스캔하게 된다(여기서 집합의 사이즈란 그 집합에 들어있는 원소 개수를 말한다). 사이즈가 k인 빈도 높은 항목들을 지금 막 구한 단계라고 하자. 이 때 이들을 이용해 사이즈가 k+1인 후보 항목들의 집합들을 먼저

〈그림 1〉 Apriori 알고리즘의 수행 과정



〈리스트 1〉 Apriori 알고리즘의 첫 번째 단계

```
// Fk : 사이즈가 k인 빈도 높은 항목 집합들
// Ck : 사이즈가 k인 후보 항목 집합들
F1 = {빈도수가 높은 항목들}
for (k=1; Fk ≠ {} ; k++) do {
    Ck+1 = Fk로부터 만들어진 새로운 후보 빈도 항목 집합
    for each 데이터베이스의 트랜잭션 t do
        t에 들어있는 Ck+1의 모든 후보 빈도 항목 집합의 카운터를 하나씩 증가시킨다.
        Fk+1 = Ck+1에 들어있는 후보들 중에서 최소 지지도 이상을 갖는 것들
    }
결과 = F1 ∪ F2 ∪ ... ∪ Fk
```

구한다. 예를 들면 {라면, 커피}와 {라면, 설탕}이 사이즈가 2인 빈도 높은 항목들 집합에 들어 있다면 이것으로부터 {라면, 커피, 설탕}이라는 사이즈가 3인 후보 항목들의 집합이 만들어진다. 이 때 이 집합의 원소로 구성된 사이즈가 2인 모든 부분집합이 사이즈 2인 빈도 높은 항목 집합들에 다 들어있는지 체크하고 만일 하나라도 없다면 후보에서 탈락시킨다. 이런 식으로 후보들을 만든 후에는 실제 데이터를 스캔해 후보들을 카운트하고 그런 후에 지지도를 만족하는 것들만 뽑아내 사이즈가 3인 후보 항목 집합을 만들어 낸다. 그 다음에는 다시 이들을 이용해 사이즈가 4인 후보들을 만들어 내고 더 이상 후보 집합을 만들지 못할 때까지 같은 과정을 반복한다.

〈리스트〉에서 Apriori 알고리즘의 첫 번째 단계를 의사 코드(pseudo code)로 나타냈다. 〈리스트〉에서 Fk는 원소의 수가 k인 빈도 높은 항목 집합들을 다 모아 놓은 것을 뜻하고 Ck는 원소의 수가 k인 후보 항목 집합들을 다 모아 놓은 것이다.

〈그림〉은 〈리스트〉의 알고리즘이 어떻게 수행되는지 나타낸 것이다. 여기서 편의상 구매 데이터 항목을 숫자로 표시했다. 그리고 최소 지지도는 75%라고 가정했다. 우선 모든 항목들에 대해 D를 스캔하면서 지지도를 계산하고 그 중에서 75%의 최소 지지도를 만족하는 항목들을 뽑아낸다. 이들은 〈그림〉에서 F1로 나타냈고 {2}, {3}, {5} 등이 있다. 그러면 F1을 이용해 다음 크기의 후보 항목들 집합 C2를 만든다. {2}, {3}, {5}로부터 만들어질 수 있는 모든 항목의 집합이 C2에 나타나 있다. 그러면 다시 D를 스캔해 이 후보항목 집합들의 지지도를 계산하고 최소 지지도를 만족하는 {2, 5}만 F2로 남게 된다. 이 때 F2에 하나의 원소만 있으므로 더 이상 그 다음 크기의 후보 항목 집합을 만들 수 없고 이 알고리즘은 여기서 멈추게 된다.

앞에서처럼 첫 번째 단계에서 얻어진 빈도 높은 모든 항목 집합으로부터 두 번째 단계에서 모든 연관 규칙을 뽑아낼 수 있다. 예를 들어 {라면, 커피, 설탕}이라는 항목 집합이 빈도수가 높다고 판명됐다면, 이것으로부터 {라면, 커피} → {설탕}, {라면, 설탕} → {커피}, {커피, 설탕} → {라면} 등과 같은 연관 규칙이 만들어지고 이것들에 대해 신뢰도를 계산한 후 최소 신뢰도를 만족하는 것만 남긴다.

는 관계형 데이터에서도 연관 규칙을 찾는 기술이 이미 개발돼 있다. 그 외에 연관 규칙 알고리즘에 관한 자세한 내용은 관련 논문을 참조하기 바람이며 이 글에서는 지면 관계상 생략하기로 한다.

연관 규칙을 사용할 수 있는 응용 분야로는 우선 백화점이나 잡화점에서 쿠폰이나 우편으로 상품 정보를 보낼 때 소비자가 산 물건들을 보고 구매할 가능성이 높은 항목들에 대해서만 상품 정보를 제공하는 것이 있다. 또 물건을 진열할 때 같이 구매할 가능성이 높은 항목들은 같이 배열해 소비자의 동선을 줄일 수 있다. 또한 웹 페이지를 디자인할 때 고객 부류에 따른 웹 페이지나 배너를 제공할 수 있다. 또한 의사가 환자에게 불필요한 검사나 치료를 해 의료를 불필요하게 줄 수 있다. 또한 의료를 불필요하게 줄 수 있다. 또한 의료를 불필요하게 줄 수 있다.



순차 패턴(Sequential Patterns)

연관 규칙은 물건을 한 번에 살 때 같이 구매한 것들을 이용해 규칙을 찾는 것인 반면, 순차 패턴 발견은 순서대로 일어난 데이터를 분석해 빈도수가 높은 순차 패턴을 찾아내는 기술을 말한다. 예를 들어 비디오테이프 대여점에서 고객의 데이터를 분석해 '포리스트 검프'를 빌려본 고객은 나중에 아폴로 13이나 캐스트 어웨이를 함께 빌려 본다는 패턴을 발견했다고 가정하자. 이러한 패턴 정보를 편의상 '(포리스트 검프) → (아폴로 13, 캐스트 어웨이)'로 나타내보자. 이 순차 패턴 정보에 따라 다른 고객 역

〈표 3〉 비디오 대여점 고객 데이터

고객 번호	구매 기록
1	((포리스트 검프), (아폴로 13, 캐스트 어웨이))
2	((포리스트 검프), (아폴로 13, 공동경비 구역))
3	((리브레터), (시월 이야기, 동감), (시월애))
4	((포리스트 검프), (캐스트 어웨이))

시 포리스트 검프를 빌려본 사람이고 나머지 두 영화를 아직 빌려 보지 않았다면 대여점을 방문했을 때 이 두 영화를 추천해 줄 수 있다. 순차 패턴의 또 다른 예를 살펴보자. 가전제품 회사에서 고객의 구매 패턴을 조사해본 결과 '(세탁기), (건조기) → ((HDTV))' 라는 패턴을 발견했다고 가정하자. 이 패턴의 의미는 세탁기를 먼저 사고, 그 다음에 건조기를 산 사람들은 나중에 HDTV를 사는 경우가 많다는 것이다.

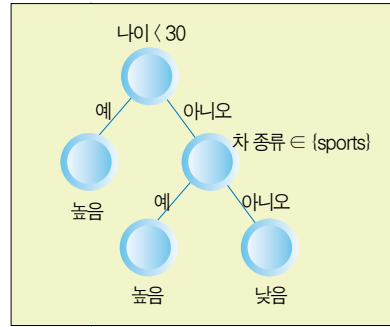
〈표 3〉은 비디오 테이프 대여점의 고객 네명의 대여 데이터다. 예를 들어, 3번 고객은 먼저 리브레터라는 영화를 대여한 후에 시월 이야기와 동감을 함께 대여했고, 그 다음에는 시월애라는 영화를 대여했다는 것을 의미한다. 〈표 3〉에서 보면 '(포리스트 검프) → (캐스트 어웨이)' 라는 패턴은 네명 중에서 두 명이 갖고 있는 패턴이므로 지지도는 50%가 되고, 만일 최소 지지도를 50%라고 했다면 앞의 데이터에서 발견할 수 있는 순차 패턴은 '(포리스트 검프) → (캐스트 어웨이)'와 '(포리스트 검프) → (아폴로 13)' 등이 있고 둘 다 지지도는 50%이다.

이 기술의 응용 사례를 살펴보자. 홈쇼핑 회사에서 소비자가 구매한 물건을 보고 다음에 살 것으로 예상되는 물건들의 쿠폰이나 카탈로그를 발송하는 데 사용할 수 있고, 학습지 회사에서는 국어 학습지를 구독하는 학생들이 그 다음에 어떤 다른 과목을 주르더 구독하는지 알아내 판매를 촉진하는 데 사용할 수도 있다. 우편 주문이나 전자상거래 사이트에서 고객이 미래에 구매할 물건을 예측하는 데 사용할 수 있고 웹 페이지 방문자들의 액세스 로그를 분석해 웹 페이지를 고객에 따라 다른 구조를 갖게 하는 데 사용할 수도 있다. 또 병원에서 진료 받은 환자들의 진료 기록을 보고 과거의 어떤 증

〈표 4〉 보험 회사 고객의 데이터

나이	차 종류	위험도
21	Family	높음
25	Sports	높음
43	Sports	높음
68	Family	낮음
32	Truck	낮음
28	Family	높음

〈그림 1〉 결정 트리(Decision Tree)



상이나 치료 과정(또는 결과)이 지금 현재 걸린 병을 유발하는 원인이었는지 찾아내는 데 이용할 수도 있다.

분류(Classification)

분류는 주어진 데이터와 각각의 데이터에 대한 클래스가 주어진 경우, 그것을 이용해 각각의 클래스를 갖는 데이터들은 어떤 특징이 있는지 분류 모델을 만들고, 새로운 데이터가 있을 때 그 데이터가 어느 클래스에 속하는지 예측하는 것을 뜻한다. 예를 들어 신용카드를 발행하는 은행의 고객 데이터에 고객의 나이, 연봉, 결혼 유무, 성별 등의 데이터와 신용 불량자인지 나타내는 클래스가 있다고 하자. 이러한 경우에 이 데이터를 결정 트리 알고리즘에 입력한다면, 신용 불량자는 나이가 23세 이하이고 연봉이 없는 사람들임을 발견한 규칙을 제공할 수 있다. 또한 미국에서는 어떤 동네든지 그 동네의 대형 슈퍼마켓에서 구매된 데이터를 사서 보면 그 동네의 아이와 젊은 여성, 노인의 비율을 대략적으로 알 수 있다고 한다. 이러한 데이터를 이용해 과거 다른 동네의 데이터와 비교해 성공했는지 실패했는지를 나타내는 클래스를 사용하면 새로운 동네에 새 슈퍼마켓을 만들려고 할 때 성공 가능성을 예측하는 데 사용할 수 있다. 또 새로운 의약품 개발했을 때 여러 부류의 사람들, 즉 연령, 인종, 성별, 체중, 키 등이 서로 다른 사람들에게 임상 실험을 한 후, 그 약품이 효능이 좋았는지, 부작용이 있었는지를 나타내는 정보를 클래스로 만들어 입력한 후 각각의 클래스의 특징을 결정 트리 기법을 사용해 만들어 의사가 약을 처방할 때 주의 깊게 사용하도록 감독할 수 있다. 또 전자상거래 사이트에서 고객들의 구매 데이터를

보고 어떤 특징이 있는 고객이 비싼 수입 명품을 구매하는지 예측하는 데도 사용할 수 있다.

이러한 분류를 위해 개발된 알고리즘으로는 결정 트리 알고리즘, 베이저안 네트(Ba

yesian Network), 신경망(Neural Network) 등이 있다. 예를 들어 〈표 4〉와 같은 보험 회사 고객의 데이터가 있고 위험도가 클래스를 나타내는 열이라 한다면, 의사 결정 트리 알고리즘은 〈그림 1〉과 같은 의사 경

정 트리를 만들어 준다. 이 트리를 루트 노드부터 살펴보면 나이가 30세 미만이면 위험도가 높고, 그 반대일 경우에는 소유하고 있는 자동차의 종류에 따라 사고를 낼 위험성이 달라짐을 나타낸다. 분류는 이처럼 각 부

군집화 알고리즘

군집화 알고리즘은 크게 파티션 알고리즘(partitional algorithm)과 계층 알고리즘(hierarchical algorithm)으로 나눌 수 있다. 전자에 해당되는 알고리즘은 K개의 모든 가능한 파티션을 모두 열거해 보고 군집화가 얼마나 잘 됐는지 나타내는 척도를 나타내는 함수 값이 가장 좋은 것으로 그룹들을 정한다. 이 때 대표적으로 많이 사용하는 척도 중의 하나는 square-error criterion인데 다음과 같은 식으로 나타낼 수 있다. 이 때 C는 각각의 그룹을 말하며, p는 각 그룹에 속한 각각의 데이터이고 m_i는 각 그룹의 점들의 평균에 해당한다. 또 ||·||는 벡터의 크기를 말한다.

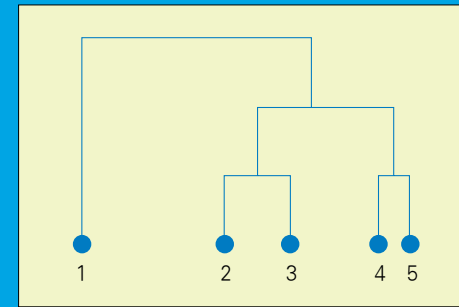
$$\sum_{i=1}^k \sum_{p \in C_i} \|p - m_i\|^2$$

숫자 속성(numeric attribute) 데이터를 군집화하는 데 쓰이는 가장 오래되고 잘 알려진 파티션 알고리즘 중에 K-평균(K-means) 군집화 알고리즘이라는 것이 있다. 이 알고리즘을 사용하려면 몇 개의 그룹으로 나누기 원하는지 K를 입력해야 한다. 그러면 알고리즘은 일단 K개의 평균점을 지정하고, 모든 데이터를 하나씩 보면서 가장 가까운 평균점에 해당되는 그룹에 할당한다. 그 후에, 다시 평균점들을 조금씩 바꾸어 나가면서 데이터를 가까운 그룹에 재할당하는 과정을 군집 상태를 나타내는 척도 함수가 더 이상 변하지 않을 때까지 반복한다. 더 이상 변하지 않게 되면 그 상태의 그룹들을 군집화의 결과로 정한다.

계층 알고리즘에는 top-down과 bottom-up 알고리즘이 있다. 그 중에서 다음과 같은 bottom-up 알고리즘을 agglomerative hierarchical clustering 알고리즘이라 한다. 이 알고리즘에서는 우선 모든 n개의 데이터가 n개의 서로 다른 그룹이라 가정된 후에 그룹간의 유사성(similarity)을 보고 가장 유사한 두 개의 그룹을 합병(merge)해 그룹 수를 줄여가는 과정을 전체 그룹 수가 K개가 될 때까지 반복함으로써 K개의 그룹을 찾아낸다. 예를 들어 〈그림〉은 1차원 데이터를 이용해 군집화하는 과정을 나타낸다.

유사성을 나타내는 함수는 두 점간의 Euclidean 거리를 쓰기로 한다고 가정하자. 〈그림〉을 보면 데이터 4번과 5번이 가장 거리가 가깝기 때문에 먼저 합친다. 이 합쳐진 그룹을 2라 부르도록 하자. 그러면 그룹은 세 개로 줄어들었다. 다시 두 개의 그룹을 합하려 할 때는 그룹 2와 4가 가장 가까우므로 합쳐서 한 개의 그룹으로 만든다. 이런 방식으로 남아있는 그룹의 개수가 K일 때까지 반복한다. 이 알고리즘의 장점 중 하나는 군집화를 마칠 때까지의 과정이 트리 구조를 가지고 있어 drill down하거나 drill up하면서 어느 단계에서 군집화를 멈추는 것이 가장 잘 된 군집화인지를 사용자가 모두 확인해 보고 가장 좋다고 생각되는 것을 선택할 수 있다는 것이다. 예를 들어 만

〈그림〉 1차원 데이터 군집화 과정



일 세 가지 그룹으로 〈그림〉의 데이터를 나누고자 한다면 맨 왼쪽 점은 한 개의 그룹으로 혼자 남고 나머지 점들은 그 다음 왼쪽부터 뿔뿔히 그룹을 지어주면 된다.

일반적으로 다차원 데이터를 군집화하기 위해 유사성을 계산할 때 여러 가지 함수를 쓸 수 있다. Euclidean 거리, Manhattan 거리 등 여러 가지가 있는데 그 중에 어떤 것을 사용해 군집화를 하느냐에 따라 다른 특성을 갖는 군집화가 이뤄진다. 자세한 내용은 참고 자료를 보기 바란다.

이러한 두 종류의 군집화 알고리즘은 컴퓨터 이론 분야에서 이미 잘 알려진 알고리즘인데, 수행 속도를 이론적으로 표현할 때 입력 데이터의 수를 n이라 하면 적어도 n²에 비례한다고 나타낼 수 있다. 하지만 n²에 비례할 경우에는 모든 데이터를 메인 메모리에 넣고 알고리즘을 수행해야만 실제 쓸 수 있게 된다. 다시 말해 데이터가 메인 메모리에 다 들어가지 못하면 수행 시간이 너무 길어 사용할 수 없다. 이러한 문제점을 극복하기 위해 BIRCH라는 알고리즘이 개발됐다. 데이터가 너무 많아 메인 메모리에 다 넣을 수 없을 경우 우선 데이터를 메인 메모리에 들어갈 수 있는 만큼 요약된 대표들을 뽑아내는 초기 군집화(pre-clustering)를 먼저 행한 후 그 요약된 데이터만 갖고 기존 K-평균 군집화 알고리즘이나 그 밖의 다른 메인 메모리용 알고리즘을 수행한다. 실제 데이터로 군집화를 수행하는 것이 아니라 전체 데이터의 분포 상태를 나타내는 요약된 정보를 가지고 군집화를 행한다. 따라서 이 과정이 다 끝난 후에는 원래 데이터에서 하나씩 보면서 이미 형성된 군집들의 특성을 보고 가장 가까운 그룹으로 배정한다.

지금까지는 주로 숫자 속성을 가진 데이터의 군집화 알고리즘을 설명했는데 백화점 고객의 구매 데이터와 같은 범주 속성(categorical attribute) 데이터에 대한 군집화 알고리즘도 여러 가지가 개발됐다. 자세한 내용은 참고 자료를 찾아보기 바란다. 대표적인 범주 속성을 위한 알고리즘으로는 ROCK 알고리즘 등이 있다.

특집 1부

류에 속하는 데이터의 특징을 찾아 새로운 데이터의 클래스를 결과로 나타내어 주는 기술을 말한다.

군집화(Clustering)

군집화 기술은 전체 데이터의 분포 상태나 패턴 등을 찾아내는 데 유용하게 이용할 수 있다. 군집화란 주어진 n개의 점을 K개의 그룹으로 나누는 것을 말한다. 분류와 다른 점은 각 클래스에 해당되는 정보가 제공되지 않는다는 것이다. 연관 규칙에 관한 부분에서 언급한 예를 들어 설명하면 주어진 여러 고객의 구매 데이터를 바탕으로 그 구매 상품의 특징에 따라 고객을 여러 그룹으로 나누는 것이라 할 수 있다. 또 모든 고객의 신상 정보를 이용해 그 유사성에 따라 그룹을 나누는 데 사용할 수도 있다. 인터넷 검색엔진 회사에서는 웹 페이지의 내용에 따라 그룹을 만드는 'categorization'에 사용할 수 있다. 군집화에는 여러 가지 알고리즘이 개발됐는데 알고리즘에 따라 다른 군집화를 만들어 낸다. 그러므로 모든 알고리즘의 특성을 잘 알고 있어야 자기 응용 분야에 맞는 것을 잘 사용할 수 있다. 또한 숫자 형태의 데이터인가 범주 형태의 데이터인가에 따라 다른 형태의 알고리즘이 존재한다.

아웃라이어 판별 (Outlier discovery)

대부분의 데이터마이닝 기술은 데이터를 나타내는 패턴에 관심을 갖고 찾아내려 한다. 하지만 아웃라이어 판별 기법은 이와 반대로 대부분의 데이터와 다른 소수 또는 일부를 찾아내는 기술이다. 이 기술은 여러 유용한 곳에서 사용할 수 있다. 예를 들면 전화 카드를 훔쳐서 사용할 경우 자신의 카드를 사용하는 대다수의 선의의 고객들과 당연히 사용 패턴이 다를 것이다. 또 훔친 신용카드를 쓰는 사람들의 사용 패턴은 자기 카드를 사용하는 고객과 다를 수밖에 없을 것이다. 또 회사나 백화점 같은 곳에서 일반 고객의 동선과 도둑의 동선은 다를 것이다. 또 시스템에 침입한 크래커들이 사용한 명령어(command)들은 정상적인 사용자들과 다를 것이다. 이러한 아웃라이어를 판별하는 데



여러 가지 기술이 이미 통계학 분야에서 사용됐는데, 여기서 사용되는 알고리즘들은 주로 데이터를 일정한 통계적 분포(statistical distribution)로 가정해 모델을 설정하고 그 모델에 따라 아웃라이어를 판단하게 된다. 하지만 많은 경우에 사용자가 자신이 이용하려는 데이터의 분포를 알고 있지 않은 경우가 더 많다. 이를 위해 데이터베이스 분야에서 distance-based outlier discovery 알고리즘들이 개발돼 있다. 자세한 내용은 참고 자료를 보기 바란다.

데이터 마이닝 기술을 사용할 때 유의할 점

데이터 마이닝 기술을 성공적으로 사용하기 위해 유의해야 할 사항은 다음과 같다. 우선 값비싼 소프트웨어보다는 작업을 수행할 팀이 더욱 중요하다는 점이다. 비즈니스, 재무, 통계학, 인공지능, 데이터베이스 등에 관해 모두 잘 알고 있어야 할 뿐 아니라 각각의 데이터 마이닝 도구에서 사용된 가정이나 장단점 같은 것을 잘 알고 사용해야 한다. 그렇지 않으면 데이터 마이닝 도구에서 만들어낸 결과에 대해 잘못된 결론을 만들 수도 있기 때

문이다. 또 사용할 데이터를 잘 이해하고 있어야 한다. 예를 들어 데이터에 빠져 있는 정보가 어떤 것인지 아는 것도 중요하고 외부로부터 얻을 수 있는 데이터들도 함께 사용해 분석하면 더욱 가치 있는 정보를 추출할 수 있다. 그리고 최근에 개발되어 많이 사용하지 않은 기술보다는 이미 사람들이 많이 사용하고 있고 그 성질이나 장단점이 잘 알려진 데이터 마이닝 알고리즘을 사용하는 것이 더 바람직하다. 잘 모르는 알고리즘을 사용하면 얻어진 결과에 대해 제대로 이해하거나 평가하기가 힘들기 때문이다.

데이터 마이닝에 관한 잘못된 인식들

데이터 마이닝에 관해 흔히 잘못 이해하고 있는 것이 몇 가지 있다. 보통 사람들은 데이터 마이닝은 데이터 웨어하우스를 구축해야만 가능하다고 생각한다. 하지만 데이터 마이닝은 디스크 파일로 된 데이터일지라도 가능하다. 대부분의 상용 소프트웨어는 텍스트 파일 형태의 데이터라 할지라도 입력해 사용할 수 있게 되어 있다. 또 어떤 사람들은 데이터 마이닝을 인공지능이나 통계학으로 단순하게 생각하는 경우가 있다. 하지만 데이터 마이닝에 두 분야의 지식이 도움이 되긴 하지만 인공지능이나 통계학이라고 간단하게 단정지어 말할 수 없다는 점을 강조하고 싶다(물론 데이터 마이닝에 통계학 지식이 전혀 필요 없는 것은 아니다). 데이

터 마이닝 기술은 무조건 도움이 되기 때문에 아무데나 사용하라고 하는 사람들이 있지만 이것도 잘못된 생각이다. 기술을 잘 이해하지 못하면서 무조건 함부로 뛰어들 경우 실패할 확률도 높아지기 때문이다. 하지만 그렇다고 데이터 마이닝 기술이 성공적으로 사용될 수 없다고 함부로 단정짓는 것도 역시 잘못된 생각이다.

데이터 마이닝, 어디로 향하고 있는가

지금까지 짧은 역사이지만 많은 유용한 데이터 마이닝 기술과 소프트웨어가 개발됐다. 하지만 이 기술들이 우리 일상생활을 정말로 얼마나 편리하게 도와줄 수 있을지는 아직 두고 봐야 할 단계에 있다. 따라서 우리 모두가 자신의 분야에서 데이터 마이닝 기술을 효과적으로 적용하기 위해 노력할 필요가 있다.

한편 인터넷과 웹의 발전은 여러 가지 웹 마이닝 문제를 우리에게 새롭게 제시하고

있다. 인터넷의 많은 웹 사이트에서는 고객들이 스스로 자신의 ID를 등록하고 자신에 관한 신상명세서를 스스로 입력한다. 그리고 여러 가지 사용 기록이 자신도 모르는 사이에 자동으로 컴퓨터에 저장된다. 이러한 많은 양의 데이터는 과거에 감히 상상할 수 없었던 것이다. 웹의 데이터는 비정형 구조(semistructured)이기 때문에 이를 이용해 유용한 정보를 추출하기가 훨씬 더 어렵다. 또한 하이퍼링크(hyper-link)를 통해 여기저기에 흩어져 있는 방대한 데이터를 잘 이용해 유용한 정보를 뽑아낼 수 있도록 하는 것도 아주 중요하다. 웹에는 너무나 많은 데이터가 있지만 또 한 편으로는 쓸모없는 데이터도 많아 원하는 정보를 찾으려고 할 때 어려움이 있다. 웹에 적용할 수 있는 웹 마이닝 기술의 개발이 현재로는 시급하면서도 앞으로 가장 매력적인 데이터 마이닝의 한 분야가 될 것이다. **✎**

정리 : 송우일 wooil@sbmedia.co.kr

주요 데이터 마이닝 솔루션과 공급 업체

IBM 인텔리전트 마이너(Intelligent Miner)

- ◆주요 기술 : 연관 규칙 의사 결정 트리, 신경망 회로, 군집화, 순차 패턴, 유사시 계열 시퀀스, 텍스트 마이닝

오라클 다윈(Darwin)

- ◆주요 기술 : 연관 규칙 의사 결정 트리, 신경망 회로, regression, 군집화, Bayesian learning, self-organizing maps, memory-based reasoning

SAS 엔터프라이즈 마이너(Enterprise Miner)

- ◆주요 기술 : 연관 규칙 의사 결정 트리, 신경망 회로, regression
- ◆장점: SAS와 뛰어난 통합 기능
- ◆다양한 샘플링 도구 제공: 랜덤 샘플링, stratified 샘플링, n-th observation 샘플링, first-n sampling and cluster sampling

SGI MineSet

- ◆주요 기술 : 연관 규칙 의사 결정 트리, regression, 군집화, simple Bayes, decision table, boosting, automatic feature selection, cross-validation
- ◆특징: SAS 파일 import/export 유틸리티 제공, 뛰어난 visualizer를 제공

SPSS 클레멘타인(Clementine)

- ◆주요 기술 : 연관 규칙 의사 결정 트리, 신경망 회로, regression, 군집화

참조 문헌

연관 규칙

- 1 Rakesh Agrawal, Tomasz Imielinski, and Arun Swami, "Database mining: A performance perspective", IEEE Transactions on Knowledge and Data Engineering, 5(6), December 1993.
- 2 Rakesh Agrawal and Ramakrishnan Srikant, "Fast algorithms for mining association rules", the VLDB Conference, Santiago, Chile, September 1994.
- 3 Rakesh Agrawal and Ramakrishnan Srikant, "Mining generalized association rules", the VLDB Conference, Zurich, Switzerland, September 1995.
- 4 Ramakrishnan Srikant and Rakesh Agrawal, "Mining generalized association rules", the VLDB Conference, Zurich, Switzerland, September 1995.
- 5 Ramakrishnan Srikant and Rakesh Agrawal, "Mining quantitative association rules in large relational tables", the ACM SIGMOD Conference on Management of Data, June 1996.

순차 패턴

- 1 Rakesh Agrawal and Ramakrishnan Srikant, "Mining sequential patterns", Int'l Conference on Data Engineering, Taipei, Taiwan, March 1995.
- 2 Minos N. Garofalakis, Rajeev Rastogi and Kyuseok Shim, "SPRINT: Sequential Pattern Mining with Regular Expression Constraints", the VLDB Conference, Edinburgh, Scotland, UK, 1999.

분류

- 1 Rajeev Rastogi and Kyuseok Shim, "PUBLIC: A decision tree classifier that integrates building and pruning", the VLDB Conference, New York City, NY, 1998
- 2 John Shafer, Rakesh Agrawal, and Manish Mehta, "SPRINT: A scalable parallel classifier for data mining", the VLDB Conference, Bombay, India, September 1996.

군집화

- 1 Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", the ACM SIGMOD Conference on Management of Data, Seattle, Washington, June 1998.
- 2 Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, "CURE: An Efficient Clustering Algorithm for Large Databases", the ACM SIGMOD Conference on Management of Data, Seattle, Washington, June 1998
- 3 Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes", the 15th International Conference on Data Engineering, Sydney, Australia, April 1999.
- 4 Tian Zheng, Raghu Ramakrishnan, and Miron, "BIRCH: An efficient data clustering method for very large databases", the ACM SIGMOD Conference on Management of Data, Montreal, Canada, June 1996.

아웃라이어 판별

- 1 Edwin M. Knorr and Raymond T. Ng, "Algorithms for mining distance-based outliers in large datasets", the VLDB Conference, New York, USA, September 1994.
- 2 Sridhar Ramaswamy, Rajeev Rastogi and Kyuseok Shim, "Efficient algorithms for mining outliers from large data sets", the ACM SIGMOD Conference on Management of Data, Dallas, TX, May 2000.

데이터 마이닝에 관한 최신 정보는 이 곳에서

데이터 마이닝 분야와 관련된 최근 기술 동향을 알고 싶다면 우선 ACM SIGKDD(Special Interest Group on Knowledge Discovery in Data and Data Mining)라는 단체에 관심을 갖기 바란다. ACM은 전산학 분야에서 가장 권위 있는 단체의 하나로 각기 다른 소분야마다 그룹이 나누어져 있고, 국제 학술회의를 열거나 뉴스레터를 제공한다. ACM SIGKDD의 홈페이지 주소는 http://www.acm.org/sigkdd/이고, 매년 2회 발행되는 뉴스 레터 SIGKDD Explorations의 홈페이지 주소는 http://www.acm.org/sigkdd/explorations/이다. 이 곳에 가면 필자가 객원 편집자(Guest Editor)로 참여했던 최근호뿐만 아니라 과월호도 온라인으로 모두 읽어볼 수 있다. 또한 'Data Mining and Knowledge Discovery'라는 국제 저널도 최근에 만들어져 새로운 기술이 소개되고 있다. 그 외에도 ACM SIGMOD, VLDB, ICDE와 같은 국제 학술회의에도 데이터 마이닝에 관련된 기술이 소개되고 있고, ACM TODS, IEEE TKDE, Information Systems 등과 같은 국제 학술지에도 데이터 마이닝 관련 기술이 소개되고 있다.