제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

네트워크 게임 환경에서의 데이터마이닝 기법에 관한 연구

김정웅*, 양해술*
*호서대학교 벤처 전문대학원
jwk@korea.com, hsyang@office.hoseo.ac.kr

A Study about Data Mining Algorithm on Network Game Environment

JeongWoong Kim* Haesool Yang*
* Dept. of Application of computer Technology, Hoseo
Graduate school of venture

요 약

게임을 구성하는 다양하고 방대한 데이터 집합을 K-Means 알고리즘을 통해 물리적 혹은 추상적 객체를 비슷한 객체군 으로 그룹화 하는 클러스트링화 하여 데이터마이닝 기법을 통한 데이터 최적화 방안을 모색한다. 이는 네트워크 환경에서의 게임의 안정성과 정형화된 패턴을 벗어나지 못하는 게임에 충분한 게임성을 높일 수 있다.

1. 서론

대규모의 데이터 내에 숨겨져 있는 고급 정보를 추출해서 의사결정, 예측, 예보에 응용하고자 하는 기법인 데이터마이닝은 데이터베이스로부터의 지식 발견(Knowledge Discovery in Database)이라 일컬어지는 데이터베이스 응용기술 분야이다.

데이터마이닝에 관한 연구는 본래 인공지능(Artificial Intelligence)의 로봇 기계 학습 등에서 시작되었으나, 본 연구는 정규화 된 게임 데이터들에 국한된 것만을 연구 대상으로 하여 종합 컨텐츠 문화 장르인 게임의 방대한 데이터를 최적화하는 방안을 데이터마이닝 기법과 그 도구인 클러스터링 개발을 위한 K-Means 알고리즘을 제시하여 해법을 찾으려 한다.

2. 데이터웨어하우스와 데이터마이닝

정보기술의 발달에 따라 기업들은 데이터베이스를 실제 업무에 있어서의 활용도를 높이기 위한 방편으로 정체되고 일관성 있게 통합된 형태로 쌓아두고자 하는 정보의 근간인 데이터웨어하우스의 구축을 시도하였다. 또한, 네트워크 환경의 발전으로 이러한 데이터들을 보다 손쉽게 접근할 수 있고 효과적으로 활용할 수 있도록 하는 연구는 계속되고 있다[1].

그러나 상업적 요구에 의해 데이터를 잘 쌓아놓은 단계를 넘어서 효과적이고 합리적인, 그러면서 신속한 전략 또는 의사 결정을 뒷받침해 줄 수 있는 의미 있는 고급 정보가 필요하게 되었다.

이러한 상황에서 이미 알려져 있고 기대했던 정보뿐만 아니라 전혀 예상하지 못했고 쉽게 드러나지도 않는 정보까지 처리할 수 있는 데이터마이닝이 등장하게 되었고 이는 네트워크, 다양한 IT정보기술 등과 함께 향후 정보산업을 이끌어갈 주체가 될 것이다.

3. 데이터마이닝

데이터마이닝은 신경망모형(Neuaral Networks Model)이나 의사 결정나무(Decision Tree)와 같은 특정 기법이 아니라, 개념적인 정보추출의 방법론이며 일련의 과정(Process)이다.

데이터마이닝의 배경이 되는 귀납적, 통계적, 기계적 학습을 살펴보면 다음과 같다. 귀납적 학습은 데이터로부터 정보의 추출이고 데이터베이스에서 같은 환경의 패턴들을 발견하려는 하나의 뷰로 분석되는 모델 형성과정이다. 유사한 객체들은 클래스로 분류되어 지고 규칙들은 보이지 않는 객체들의 클래스를 예상할 수 있게 공식화되었다. 통계적 학습은 이론적 기초는 있지만 통계학의 결과는 예상이 어렵고, 그들이 어디서 어떻게 데이터를 분석할지 사용자가 안내를 요구하는 만큼 해석하기가 어렵다. SAS와 SPSS, 그리고 S-PLUS같은 통계적 분석 시스템은 특별한 패턴의 감지와 선형(Linear)모델 같은 분석의 역할을 하고, 데이터마이닝은 분석이 아닌 결과를 중심으로 더욱 더 가시적인 분석에 적극 활용을 할 수 있다. 기계적 학습은 학습과정의 자동화이고 학습은 환경적 상태와 변화의 관측에 기초한 규칙을 만드는 것이다. 기계학습은 이전 예제와 그들의 결과를 검사하고 이것을 어떻게 재생산하는지 배우며 새로운 경우에 관한 일반화를 만든다.

3. 데이터마이닝을 이용한 게임 데이터 최적화

게임성을 추구하는 게임의 성격상 다양한 형태의 주체 그리고 그들의 활동, 그것들을 표현하는 배경 등 게임은 방대한 양의 데이터를 요구하고 있다. 또한 게임 진행을 위한 데이터 처리 및 데이터베이스 관리기능 등을 수행해야 하며 기본적으로 많은 사용자가 동시에 서버에 접속하여 진행하는 네트워크 게임에서는 특성상 데이터의 최적화 처리가 중요하다.

데이터마이닝에서 개발된 많은 기술과 도구는 게임 데이터에서 정보나 지식을 발견하고 추출한다. 데이터 분석도구는 게임 데이터 및 정보처리의 여러 분야에 도움이 되는 이미 개발된 기술(NLP)의 Tool Set 으로 데이터마이닝 툴에서 제공하고 있는 기능들을 참고하여 일반적으로 다음과 같이 정리된다[10][11].

첫째, 제작된 게임의 데이터를 자동으로 찾아내는 데이터 식별도구, 둘째, 게임 데이터를 미리 정의된 카테고리나 분류 등을 자동으로 지정해주는 데이터 분류도구, 셋째, 데이터의 항목을 자동으로 인식하는 특성추출도구, 넷째, 유사한 데이터집합을 자동으로 그룹이나 '클러스터'로 나눌 수 있는 클러스트링 도구, 다섯째, 데이터를 분석하여 요약정보를 추출하는 요약도구 등이 있다.

4. 클러스터링과 K-means 알고리즘의 적용

데이터마이닝을 결과로 표현할 수 있는 기반 지식에는 연관규칙, 요약규칙, 클러스터링 등 여러 가지가 있다. 본 연구에서는 이 데이터마이닝의 지식 기법 중에서 클러스터링 기법을 통하여 게임 데이터를 어떻게 연관하고, 가공하고 추출하는지를 제시하고, 클러스터링에 대한 개발기법 중의 하나인 K-Means 알고리즘을 제시하여 해법을 찾으려 한다. 우선 여러 가지 기법들에 대해서 약술하고, 클러스터링과 K-Means에 대해서 논하겠다[7].

데이터 분류란 데이터베이스 내의 객체의 집합에 대하여 그 안에 내재하는 공통 특성을 뽑아내어 이 객체들을 서로 다른 클래스로 그룹화 하는 작업을 말한다. 이 작업은 먼저 트레이닝 집합을 분석하여 각 클래스별로 정확한 묘사 또는 모델을 생성해 내어 이를 이용해서 목표 데이터를 처리한다. 그리고 클러스터링이란 물리적 혹은 추상적 객체를 비슷한 객체군 으로 그룹화 하는 과정이다. 이 때 유사성 때문에 함께 모여진 개체의 집합을 클러스터링이라고 한다. 클러스터링 작업은 먼저 필수 객체들이 셋으로 모여지고 이로부터 일련의 규칙이 유도된다[9].

4.1 클러스터링(Clustering)

클러스터링은 관련된 객체의 부분 집합을 발견하고 이 부분 집합의 각각을 기술하는 묘사를 발견해 나가는 방법이다. 일반적으로 클러스터링이라 하면, 두 개 이상의 워크스테이션이나 시스템을 LAN 등의 물리적인 연동 미디어를 사용하여 마치 하나의 시스템인 것처럼 보이게 하는 방식이다. 이 방식은 디스크에서의 병목현상에 의해 확장성에 어느 정도 제약을 받는 구조이기는 하나 클러스터 단위의 소규모 확장을 용이하게 함과 더불어 가용성을 높일 수 있는 유효한 방식으로 평가된다. 이러한 클러스터링 병렬기술은 하드웨어를 이중화함으로써 높은 신뢰성을 실현할 수 있다. 또한 복수의 클러스터의 구성에 의하여 클러스터에 장애가 발생한 클러스트는 시스템 가동 중에 보수할 수 있기 때문에 24시간 365일 연속운전이 가능하게 된다.

이와 같이 클러스터링 방식은 CPU수 증가에 따른 성능증가의 어려움이 단점으로 나타나는 SMP방식(Symmetric Multi Processing : 병렬 컴퓨터 기술에서 모든 프로세스가 메모리와 디스크를 공유하는 방식)과 상대적으로 애플리케이션 등의 미비 등으로 인해 본격적인 상용서버로 자리 매김하지 못하고 있

는 것이다. 클러스터링은 일반적으로 2대에서 많게는 8대까지의 워크스테이션이나 시스템을 LAN이나 FDDI 등의 미디어를 사용, 상호 연동해 사용한다.

4.2 클러스터링 방식에 대한 계층적 최적화 알고리즘과 구현

클러스터링 방식에 이용되고 있는 알고리즘으로서는 계층적 최적화 알고리즘이 중용되고 있다. 본 연구에는 4.2.1절과 같은 계층적 최적화 알고리즘을 4.2.2절과 같이 C-언어로 구현하여 나타내었다.

4.2.1 계층적 최적화 알고리즘

데이터마이닝에 적용되는 클러스터링 알고리즘은 비교학습의 특징을 가지면서, 정해지지 않은 다량의 게임 데이터 집합에 대해 데이터베이스의 제한적인 요소들을 얼마나 잘 충족시키는가가 중요한 요소로 다뤄진다. 따라서 본 연구에서의 최상의 알고리즘 선정 기준으로 1) 규모 확장성(scalability), 2) 복잡도(complexity), 3) 파라미터 개수(human interaction 정도)에 따라 선정하게 되었으며, 클러스터링의 접근 방식에 따라 분리해서 선정하였다. 다음 <표1>은 계층적 클러스터링에 관한 최적 알고리즘을 표로서 나타낸 것이다.

<표1> 계층적 클러스터링의 최적알고리즘

	Scalability	Complexity	Parameter 개수	Best
PAM	S(100)		1	
CLARA	S(500)		1	
CLARANS	S(1000)		2	
BIRCH	L(100,000)		2	
CURE	L(500,000)	O (logS*S^2)	5	0
ROCK	S(8,000)		3	

K: cluster 수, N: data 수, S: sample data 수
m_a: neighbor의 평균값, m_m: neighbor의 최대값

계층 기반 접근방식에서는 50만개 이상의 입력 데이터를 가질 수 있으며 선택된 샘플 크기 S에 대해 사전 클러스터링을 수행함으로써 O(logS*S^2)의 좋은 복잡도를 가지는 CURE를, 밀도 기반에서는 50만개 이상의 입력 데이터와 O(N*logN)의 복잡도를 가지는 DBSCAN과 DBCLASD, OPTICS 중에서 인간 개입이 전혀 필요하지 않은 DBCLASD를 선정하게 되었다. 이를 정리하면 <표2>와 같이 밀도를 기반으로 하는 최적 클러스터링 알고리즘을 나타낼 수 있다.

4.2.2 계층적 최적화 알고리즘의 구현

```
#include<math.h> #include<stdio.h>
#define pm pow(2,32) #define k 500000
#define m 2
long ix=1124443767;
double
  ran()(ix=ix*65539+135745;return(ix/pm+0.5))
void normal(n1,n2) double *n1,*n2;
double v1,v2,r2;
do{
  v1=ran()** 2 -1;v2=ran()
  *2 -1;r2=(v1*)v1+v2+v2;
  }while(r2>1){
    n1=(v1*)sqrt(((-2*)log(r2))/r2);
    n2=(v2*)sqrt(((-2*)log(r2))/r2);
    }
main(){
  int n=100,i,kk,j,nn[10],md,kd[100],f,i,kr,k,
    kr[100];
  double xm[m][10],sumx[m][10],x[m][100]
    ,dis,mdis,n1,n2;
  double sum=0,msum;
  for(j=0;j<m;j++)
```

<표2> 밀도기반 클러스터링의 최적알고리즘

	DBSCAN	DBCLASD	DENCLUE	OPTICS
Scalability	L(500,000)	L(500,000)	L(100,000)	L(500,000)
Complexity	O(N*logN)	O(N*logN)	O(log||D||)	O(N*logN)
Parameter개수	2	필요 없음	2	2
Best		0		

```
for(j=0;i<n;i+=2){
  normal(sn1,sn2); x[j][i]=n1;x[j][i+1]=2; }
msum=9999;
for(l=1;1k=1000;1++){          /*k-means*/
  for(i=0;i<n;i++)kkr[j]=0;
  for(kk=1;kk<=k;kk++){ kr=ran()*n;
    if(kkr[kr]==1)(kk--;continue;) kkr[kr]=1;
    for(j=0;j<m;j++){
      xm[j][kk]=x[j][kr];
      sumx[j][kk]=0;  }
    nn[kk]=0;  }
  for(i=0;1<n;i++){ mids=9999;
    for(kk=1;kk<=k;kk++){ dis=0;
      for(j=0;j<m;j++){
      dis+=(x[j][i]-xm[j][kk])*(x[j][i]-xm[j][i]);
      if(dis<=mdis){md=kk;mdis=dis;  } }
  kd[i]=md; nn[md]++;
  for(j=0;j<m;j++)xm[j][kk]=sumx[j][kk]/nn[kk];
      while(f==1) E=0;
      for(i=0;i<n;i++){ mdis=9999;
        for(kk=1;kk<=k;kk++){ dis=0;
          for(j=0;j<m;j++){
       dis+=(x[j][i]-xm[j][kk])*(x[j][i]-xm[j][i]);
  if(dis<mdis){md=kk;mdis=dis;} }
  if (md!=kd[i]{ f=1;nn[md]++;nn[kd[I]--;
      for(j=0;j<m;j++){
      sumx[j][md]+=x[j][i];
      sumx[j][kd][i]-=x[j][i];  }  kd[i]=md;  }
      for(kk=1;kk<=k;kk++){
for(f=0;j<m;j++) xm[j][kk]=sumx[j][kk]/nn[kk];{
  sum=0;  for(i=0;i<n;i++){
  for(j=0;j<m;j++){
  sum+=(x[j][i]-xm[j][kd[i]*(x[j][i]-xm[j][kd[i]]
  printf("%5d %10.4f/n",1,sum);
  if(sum<msum)msum=sum;
  printf("%10.4f/n",msum);
```

<그림1> 계층적 최적화 알고리즘

5. 결론

데이터웨어하우스가 데이터들을 보다 손쉽게 접근할 수 있고 효과적으로 활용할 수 있는 한계점을 보인 반면, 데이터마이닝은 개발된 많은 기술과 도구를 이용하여 데이터웨어하우스에서 정보나 지식을 발견하고 추출할 수 있다.

다양하고 정형화 되지 않은 게임을 위한 가상현실, 3차원 그래픽을 이용한 게임이 속속 등장하는 현 시점에 게임을 표현하는 데이터를 최적화 하는 방안을 데이터마이닝 기법에서 찾을 수 있다.

본 연구에서는 이러한 데이터마이닝의 개념과 간단한 활용 기법을 K-means 알고리즘을 이용하여 클러스터링 도구를 다루는 것으로 고찰하였다.

결론적으로, 네트워크 환경에서 데이터베이스 구축은 게임 응용 데이터베이스 분야의 가장 핵심이 될 것이고, 그 중 데이터마이닝 기법은 게임 분야에 있어서 총체적으로 필요하게 될 것이다.

다만 데이터 추출을 위한 데이터마이닝의 혁신적 툴 개발이 선행되어야 한다.

참고문헌

[1] Two Crows Corporation. introduction to data mining and knowledge discovery. Two Crows, 1999.

[2] http://cs.sungshin.ac.kr/~de/

[3] http://members.xoom.com/khmin/db/html

[4] 조경산, "컴퓨터 네트워크와 인터넷", 도서출판 그린, 1998.

[5] 유황빈 외 1, "데이터통신", 정익사, 1996.

[6] http://dwserver.hit.co.kr/int.htm

[7] Michael J.A.Berry and Gorden Linoff. Data Mining, Techniques. John Wiley &Sons, 1997.

[8] 윤재관외 2, "공간데이터마이닝을 위한 객체 관리 시스템", 건국대, 1998.

[9] 나영민, 최병갑, "데이터마이닝을 위한 지식기반 트리분류기", 데이터베이스연구회지, 12권 4호, 1996.

[10] Dorre, J., P. Gerstl, R. Seiffert, "Text Mining: Finding Nuggets in Mountains of Textual Data", in Proceedings of the Fifth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, 1999.

[11] Lee, H-Y, "Text Mining-Knowledge Discovery from Text", Trend in Knowledge Discovery from Databases, 29th June 1999.