論文
2-3-5

교통 정보 데이터베이스를 이용한 마이닝
(Mining using Traffic Information Database)

이 기 성* 박 종 천** 김 광 휘***
(Gi-Sung Lee) (Jong-Cheon Park) (Kwang-Huy Kim)

요 약

차량이 증가함에 따라, 도로 교통은 혼잡하게된다. 교통 혼잡을 통계적 분석을 이용하여 예측할 수 있다면 도로 교통에 상당히 개선될 것이다. 본 논문은 고속도로의 속도에 영향을 주는 요소를 분석하여 상호관련성을 조사한다. 이를 수행하기 위해 고속 도로 교통에 대한 데이터베이스를 구축하며, 도로 교통 데이터베이스에다 설정된 가설을 적용하고, 다양한 데이터 마이닝 연산을 사용하여 결과를 도출한다.

ABSTRACT

According to the increasing of the cars, road traffic confused. If we estimate the traffic confusion use of statistical research, road traffic improved considerably.

This paper analysis element that affected a expressway speed and investigate the mutual relation. For the accomplish it, we construct the DB for traffic of a expressway and applied a hypothesis to road traffic DB, we obtain the results from a various method with Data Mining operation.

1. 서론

현재 우리 나라는 많은 교통 체증을 겪고 있다. 또한 교통 체증으로 인해 차의 평균 시속이 해마다 줄고 있다. 또한, 우천 시나 눈이 내려 도로의 사정이 좋지 않을 때는 차의 평균 시속이 현저히 감소된다. 이러한 도로의 상황을 단순한 통계나 관측을 이용하여 일반 운전자에게 숙지할 수 있는 정보로 가공하기는 많은 어려움이 있다. 하지만, 특정 도로의 교통정보를 특정 주기로 데이터베이스에 구축하여 원시자료를 작성하고, 그 데이터를 이용해 가설을 설립하고, 가설에 대해 마이닝의 다양한 연산(클러스트링, 연관화 등등)을 적용하면 데이터의 연관관계나 분포, 밀접성들의 결과를 쉽게 도출하여 자동차의 속도에 영향을 받는 속성들을 유추하여 분석할 수 있다[2,6,8]. 즉, 도로는 많은 속성들을 가지고 있다.

예를 들어 날씨, 도로표면상태, 도로공사, 예기치 않은 교통사고 등등이 있다. 이러한 속성들은 서로 독립적이지 않고 하나의 속성값이 바뀌면 다른 속성의 값이 바뀌는 경우가 생길 수도 있고 아닐 수도 있다.

본 논문은 이러한 도로에 대한 속성들간의 관계를 유추하기 위해 도로에 대한 교통 정보 데이터베이스를 구축하며, 가설을 설립하고, 데이터의 연관관계와 속성을 유추하여 속도에 영향을 주는 요소들을 도출한다. 또한 방대한 데이터의 자료로 인한 오차율을 막기 위해 많은 도로 중 고속도로에 대한 교통 정보 데이터를 이용한다. 논문의 구조는 2 장에서는 도로 교통 데이터베이스의 구조와 특성에 대해 기술하고, 3장은 전체 시스템의 구조와 전처리에 대해 설명한다.

* 정회원 : 숭실대학교 컴퓨터학과 박사수료
** 정회원 : 대전기능대학 멀티미디어과 전임강사
*** 정회원 : 우송정보대학 교수

논문접수 : 2001. 3. 10.
심사완료 : 2001. 3. 30.

4장은 데이터 마이닝을 이용한 분석 및 결과를 도출하며, 5장은 결론 및 향후계획을 제시한다.

2. 관련 연구

2.1 데이터마이닝

최근 수년동안 학계, 연구계, 산업계에서 데이터 마이닝에 대한 연구가 이루어져 왔다. 그간 제안된 다양한 데이터 마이닝 기법들은 어떤 형태의 지식을 탐사하고자 하는가, 어떤 종류의 데이터베이스에 적용될 수 있는가, 어떤 분야의 기술에 바탕을 두고 있는가 등의 기준에 의거하여 아래와 같이 분류한다[2,6,8].

2.1.1 탐사될 지식의 형태에 따른 분류

(1) 특성화(characterization)
데이터 집합의 일반적 특성을 분석하는 것으로 일반화 및 세분화 과정에 의한 자료 요약 과정을 거쳐 특성 규칙을 발견

(2) 분류화(classification)
다른 클래스에 대한 차별적인 특성을 도출한다. 이와 같은 차별적인 특성은 소속 클래스를 알 수 없는 미지의 객체가 있을 때, 그 소속 클래스를 결정하는데 활용된다.

(3) 군집화(clustering)
유사한 특성을 갖는 데이터들을 묶어주는 것이다. 인공지능 분야에서 분류는 감독학습암에 반해 클러스터링은 비감독 학습으로 불린다. 감독 학습이란 감독자가 자료를 집단별로 구분해 놓고 분류기준은 컴퓨터 프로그램이 학습에 의하여 발견하도록 하는 방법이다. 비감독 학습은 감독이 없이 컴퓨터 프로그램 스스로가 자료집단의 유사성을 바탕으로 집단을 나누어 나가는 방식이다.

(4) 연관화(association)
여러 개의 트랜잭션들 중에서 동시 발생하는 트랜잭션의 연관관계를 발견하는 것이다. 규칙 발견에 사용한 측정값은 연관성의 신뢰요인으로 사용된다.

(5) 경향분석(trend analysis)
시계열 데이터(주식, 물가, 판매량, 과학적 실험 데이터)들이 시간 축으로 변하는 전개 과정을 특성화하여 동적으로 변화하는 데이터의 분석을 수행한다.

(6) 패턴분석(pattern analysis)
대용량 데이터베이스 내의 명시된 패턴을 찾는 것이다.

2.2 GeoMiner

GeoMiner 시스템은 캐나다의 Simon Fraser 대학에서 관계형 데이터 마이닝 시스템인 DBMiner를 확장하여 공간 데이터를 처리할 수 있도록 개발하고 있는 시스템이다. GeoMiner의 기본이 되는 DBMiner는 데이터 마이닝과 데이터 웨어하우스 기술의 결합으로서 개발되었고, 관계형 데이터 마이닝을 위한 데이터 큐브 구축과 처리, 애트리뷰트-지향 유도, 다중 레벨 조합 분석, 통계적 데이터 분석, 기계 학습등을 포함호고 있다. GeoMiner는 질의어로서 GMQL을 제공하고 데이터 마이닝 결과를 테이블, 차트, 지도 등의 형태로 출력하기 위한 대화식 및 그래픽 사용자 인터페이스도 지원한다. 또한 , 공간 데이터의 처리를 위해 MapInfo Professional 4.1 GIS를 사용하며, 비공간 데이터, 공간 데이터, 개념 개층을 저장하는 데이터베이스를 관리한다[9].

3. 데이터베이스 구조

마이닝에 사용할 데이터베이스는 현재 도로교통망 정보서비스에서 사용하고 있는 데이터베이스로서 DBMS로는 오라클 8(Oracle 8)을 사용한다. 구축된 데이터베이스 시스템의 특성은 다음과 같다.

① 원시 Database 개요
- 특정 기업의 고속도로 정보 서비스를 위한 데이터베이스를 사용.

- 데이터베이스에는 월요일부터 일요일까지의 일주일 분량의 정보가 저장.
- 5분 단위로 새로운 정보가 추가
- 전국 20개의 고속도로 중 경부고속도로(상/하행)만 추출하여 데이터베이스를 새롭게 구성.
- 경부고속도로는 56개의 구간으로 분리되어 있고, 이중 29개의 구간을 중점적으로 사용.
- 각 구간은 인터체인지(IC), 분기점(JC), 톨게이트(TG), 휴게소를 기준으로 구분.
- DBMS는 Oracle8를 사용.

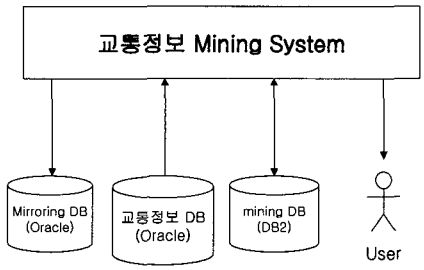
② 마이닝에 사용된 테이블 구조
- ddate : 날짜
- ttime : 시간
- link_id : 구간 id
- from_node : 시작지점 id
- to_node : 도착지점 id
- congestion_grade : 일반도로 상태 id
- speed : 일반도로 속도 (km/h)
- travel_time : 일반도로 소요시간 (초)
- bus_congestion_grade : 버스전용도로 상태 id
- bus_speed : 버스전용도로 속도 (km/h)
- bus_travel_time : 버스전용도로 소요시간 (초)
- suspension : 차단통제 정보 id
- announcement : 공지사항 id
- weather : 도로기상 id
- queue_length : 지체길이 (m)

* id로 표기되는 것은 별도의 table이 존재하기 때문에 비교하여 확인.

4. 시스템 구조도

4.1 개념도

우리의 마이닝 작업은 오라클로 구축된 데이터베이스로부터 작업하기 용이한 별도의 시스템에 같은 데이터를 미러링(mirroring)하여 작업하며, 또한 마이닝에 사용할 도구인 인텔리전트 마이너(IM)을 사용하기 위하여 DB2 DBMS에서 이식하여 작업한다. 개념도의 대략은 아래 그림과 같다[1,3,5].



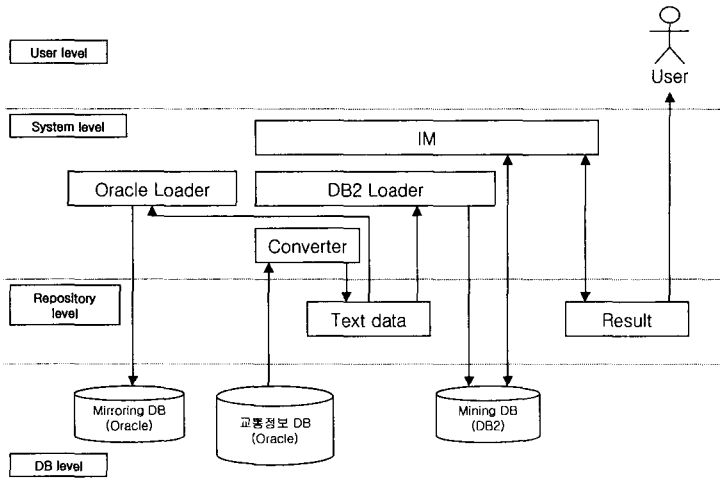
③ 자료 예제

```
SQL> select * from tsdgts.highway_value where link_id like 'K%';

DDATE    TTIM LINK_ID  FROM_NOD TO_NODE  C     SPEED TRAVEL_TIME B BUS_SPEED BUS_TRAVEL_TIME SUSP
-------- ---- -------- -------- -------- - --------- ----------- - --------- --------------- ----
ANNO WEAT QUEUE_LENGTH
---- ---- ------------
20000508 1655 KDT10010 KKN10001 KKN10002 1       100         116 0         0               0 0000
G102 E109            0

20000508 1655 KDT10020 KKN10002 KKN10003 2        68         126 0         0               0 0000
G103 E109            0

20000508 1655 KDT10030 KKN10003 KKN10004 1        90         101 0         0               0 0000
G103 E109            0

20000508 1655 KDT10040 KKN10004 KPN10005 1        94         310 0         0               0 0000
G102 E109            0

20000508 1655 KDT10050 KPN10005 KKN10006 1       100          41 0         0               0 0000
G102 E109            0

20000508 1655 KDT10060 KKN10006 KKN10007 1        99         131 0         0               0 0000
G102 E109            0

20000508 1655 KDT10070 KKN10007 KKN10008 1        95         169 0         0               0 0000
```

4.2 상세도

우리의 마이닝 시스템은 크게 위로부터 User level, System level, repository level, DB level로 이루어진다. 즉, 우리의 마이닝 시스템은 데이터베이스 레벨의 교통정보 DB로부터 사용자가 알기 쉬운 User level로 결과를 추출하는 시스템이다. 원시 자료로서 구축되어 있는 교통정보 DB는 매 5분마다 빈번하게 갱신되므로 무척 느리고, 사실상 다른 작업을 전혀 할 수 없는 상황이므로 우리는 같은 내용으로 다른 시스템에 데이터베이스를 이식하여야 한다.

따라서 이식할 시스템으로는 두 시스템이 필요하며 하나는 단순 DB작업을 할 수 있는 시스템이고, 다른 하나는 마이닝을 위한 DB작업을 할 수 있는 시스템이다. 데이터베이스 이식을 위하여는 기존 데이터를 받아서 이식 가능한 데이터로의 변형이 필요하므로 우리는 converter 역할을 할 수 있는 프로그램을 자바(java)언어를 이용하여 작성하였다[10]. converter에 의하여 작성된 결과는 오라클과 DB2 모두에게 적합한 형태를 가지므로 그대로 이식이 가능하다.



4.3 전처리

전처리는 원시 자료를 데이터베이스에 로드해서, 데이터베이스에서 작업을 수행하기 위한 형태로 변경시켜 주는 작업을 수행한다. 전처리 작업 모듈은 프로그램 언어 중 하나인 java를 이용하여 구축하였다. 전처리 작업을 마친 후의 데이터는 다음과 같다.

모든 문자열은 앞과 뒤에 큰따옴표(double quotation)로 표시되며 숫자형 데이터는 아무런 표시없이 나타난다. 각 컬럼간의 구분자는 쉼표(,)로 구분되며, 데이터베이스 테이블내에서의 로우(row)는 캐리지 리턴(carriage return)으로 표시된다. 아래는 전처리 후의 데이터 일부를 보인 것이다.

```
"20000508","1655","KDT10010","KKN10001","KN10002","1",100,116,"0",,0,"0000","G102","E109",0
"20000508","1655","KDT10020","KKN10002","KN10003","2",68,126,"0",,0,"0000","G103","E109",0
"20000508","1655","KDT10030","KKN10003","KN10004","1",90,101,"0",,0,"0000","G103","E109",0
"20000508","1655","KDT10040","KKN10004","PN10005","1",94,310,"0",,0,"0000","G102","E109",0
"20000508","1655","KDT10050","KPN10005","KN10006","1",100,41,"0",,0,"0000","G102","E109",0
"20000508","1655","KDT10060","KKN10006","KN10007","1",99,131,"0",,0,"0000","G102","E109",0
"20000508","1655","KDT10070","KKN10007","KN10008","1",95,169,"0",,0,"0000","G102","E109",0
"20000508","1655","KDT10080","KKN10008","YN10009","1",95,144,"0",,0,"0000","G102","E109",0
```

5. 데이터 마이닝을 이용한 분석 및 결과

가설 1. 속도가 가장 빠른 시간대는 언제인가?

고속도로에서 운전을 하다보면 어느 시간대에서는 차량 통행량이 많아 평균 속도가 늦은 경우가 있다. 그래서 전반적으로 어느 시간대에서 차량의 속도가 많이 늦어지는가를 알아서 그 시간대를 피하면 보다 효율적인 운전을 할 수 있을 것이라고 본다. 가설을 검증하는 방법은 아래의 방법을 이용하여 클러스터링을 수행하였다.

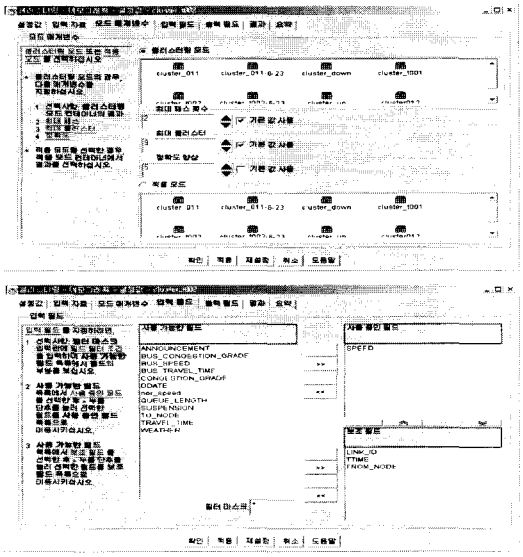
1) 상행선, 하행선의 구분없이 양방향의 차선에 대해 속도에 대해 자료들을 클러스터링한다.
2) 하행선의 한구간의 자료를 뽑아서 클러스터링 작업으로 시간 분포의 특성을 보았다.
3) 자료를 두 개로 나누어서 상행선, 하행선 부분으로 나누어서 속도에 대해 클러스터링.

- clustering 1:

상행선, 하행선의 구분 없이 양방향의 차선에 대한 속도에 대한 클러스터링을 수행한다.

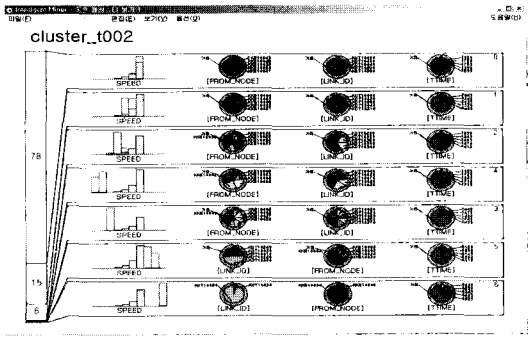
방법:

1. roadhist라는 table을 생성한다.
2. 전처리과정을 거친 자료를 테이블에 올린다.
3. 자료들 중에서 하루의 자료의 양이 많이 차이가 나는 것을 제거한다.
4. 아래의 방법으로 클러스터링을 과정을 수행.



많은 필드 중에서 클러스터링을 작업을 위한 부분은 speed필드를 이용하였다. 보조자료는 LINK_ID와 시간대인 TTIME을 넣었으며, 정확한 위치를 알기위해 FROM_NODE를 넣었다.

- 결과 화면:



- 결과 고찰

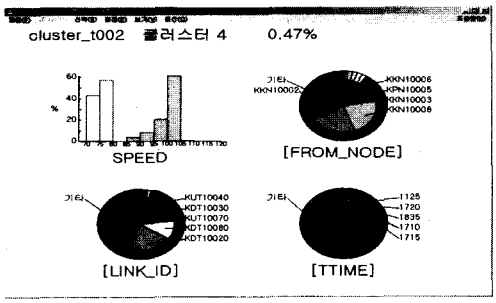
고속도로 상태의 도로는 거의 모든 자료가 독립적인 성격을 갖기 때문에 서로의 연관성을 찾는 것은 힘들다는 것을 위의 자료에서 알 수가 있다. 이유는 기타에 해당하는 자료가 많다는 것으로 알 수 있다. 이 자료 중에서도 양은 작지만 아주 미세한 부분에 의해 고속도로의 상태를 움직일 것이라고 본다.

위 자료에서는 전반적으로 평균속도를 가지는 집합이 많은 부분을 차지하다는 것을 알 수 있다. 그리고 다음으로 잡히는 집합은 저속도 집합이 많이 잡히고 마지막으로 작은 집합은 과속을 하는 집합이다. 속도에 의해서 군집합의 크기의 순위를 보면

- 정속도를 따르는 것
- 정속도에는 미치지는 못하지만 약간 정체를 보이려는 것
- 약 30-40K의 속도를 나타내는 것
- 거의 정체를 나타내는 것
- 정속도보다 약간 높은 속도를 내는 것
- 과속을 하는 것

과 같음을 알 수 있다.

정속도를 따르는 집합은 시간대에 대한 분석을 하면 전체적인 관계에서 기타의 비중이 그 집합에서의 비중과 거의 같기 때문에 시간대가 관계가 없다고 볼 수 있다. 그러나 저속도 집합을 보게 되면 약간의 차이를 보이게 된다. 아래는 클러스터4인 저속도의 집합을 확대해서 본 것이다.



(KKN10008 : 죽전휴게소, KKN10002 : 반포IC, KKN10003 : 서초IC, KKN10005 : 판교JC, KKN10006 : 판교IC)

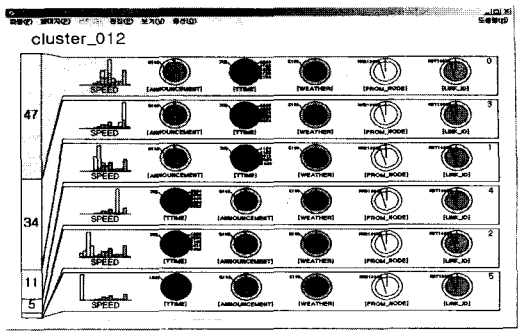
저속도를 나타내는 것은 오후 5, 6시간대와 오전 11시간대가 많이 나타난다. 그리고 구간을 보면 하행선에서 많이 잡히는 것을 알 수가 있다.

위의 자료는 LINK_ID는 FROM_NODE필드를 가지고 있는 것으로 서로의 자료의 형태는 같은 것이다. 따라서 가장 많이 지체가 되는 구간은 하행선 부분이며 일상적으로 정체가 자주되는 서초와 반포IC, 판교JC 부분이다. 상행선 부분은 죽전휴게소와 판교IC부분이다. 여기서 판교 부분이 많이 나왔다는 것은 위 시간대에서 판교 부분은 언제나 속도가 느리다는 것이다.

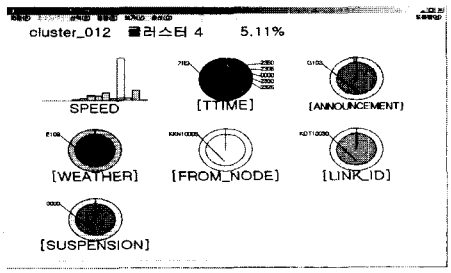
- clustering 2:
  하행선의 한 구간의 자료를 뽑아서 시간 분포의 특성을 보았다.

  - 방법: 1. 위 clustering 1의 방법과 같다. 단지 하나의 구간을 오라클에서 뽑아서 따로 자료를 만들었다. 구간은 하행선인 서초IC에서 양재IC 사이의 구간이다.

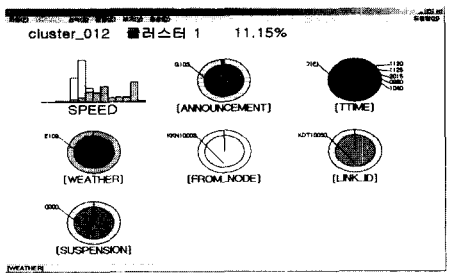
- 결과 화면:



- 결과 고찰
  전체적인 자료는 클러스터링1의 자료와 거의 같다. 여기서 다른 점은 저속를 이루는 집합들 사이에 정속도를 이루는 구간이 있다는 것이다. (클러스터 4)



정속도를 이루는 시간대는 저녁 11시경임을 알 수 있다. 그리고 정속도보다 약간 빠른 집합은 새벽 2시경임을 알 수 있다. 위의 자료에서 이 구간은 저녁 11시경에는 속도가 정속도를 이루기 시작함을 알 수가 있다.



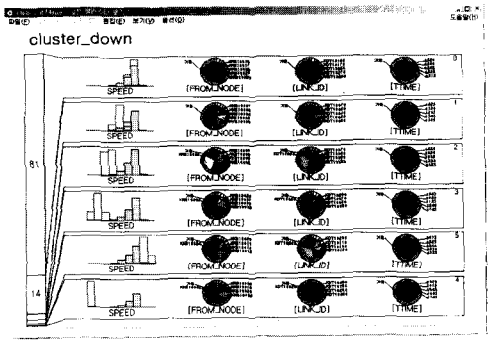
저속도를 이루는 클러스터1이 알려주는 것은 오전에 9시경부터 속도가 느려지기 시작한다는 것을 볼 수 있다.

이외의 날씨와 공지사항의 필드는 속도에는 아무런 관계가 없다는 것을 알 수 있다.

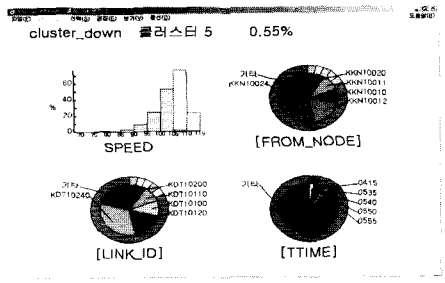
- *clustering 3:*
  하행선 부분으로 나누어서 속도에 대해 클러스터링.

  - 방법: 1. 위 clustering 1의 방법과 같다. 자료는 전체 자료는 많은 시간을 소비하는 관계로 서울에서 대전사이의 경부고속도로구간을 추출하여 실험하였다.

- 결과 화면:

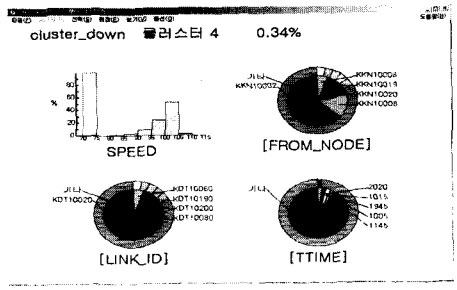


- 결과 고찰
  위의 자료를 보면 시간과 속도와의 관계를 가지는 것은 아주 극단적으로 속도가 빠르거나 느린 경우라는 것을 알 수가 있다. 정속도를 이루는 부분과 약간 속도가 느린 부분은 시간과 관계가 없다. 그러나 정체를 나타내는 부분이 나오는 부분부터는 시간과의 관계가 생기는 것을 알 수가 있다.



(KKN10024 : 청주IC, KKN10012 : 기흥IC, KKN10010 : 수원IC, KKN10011 : 기흥휴게소, KKN10020 : 천안IC)

먼저 아주 빠른 속도를 나타내는 집합을 보면 시간은 새벽에 속도가 아주 빠르게 나타나는 것을 알 수 있다. 기흥부근에는 새벽에 아주 빠른 속도로 지나 갈 수 있다는 것을 알 수 있다.



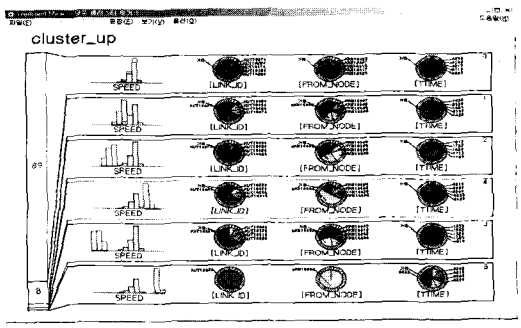
(KKN10002 : 반포IC, KKN10008 : 죽전IC, KKN10020 : 천안IC, KKN10019 : 천안삼거리휴게소, KKN10006 : 판교IC)

가장 저속도를 이루는 집합에서 시간대는 거의 오전 10에서 11시경과 저녁 7시, 8시대임을 알 수가 있다.

- clustering 4:
  상행선 부분으로 나누어서 속도에 대해 클러스터링.

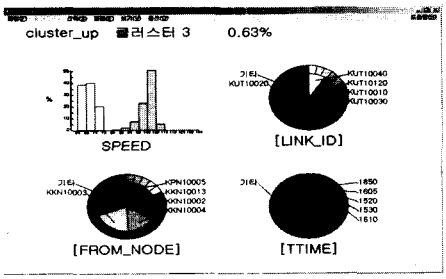
  - 방법: 1. 위 clustering 1의 방법과 같다. 자료는 전체 자료는 많은 시간을 소비하는 관계로 서울에서 대전사이의 경부고속도로구간을 추출하였다.

- 결과 화면:



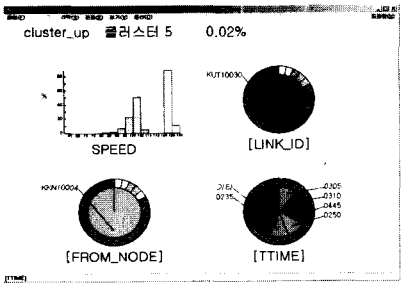
▪ 결과 고찰

위의 자료는 보면 하행선의 경우와 같은 결과가 나오고 있다. 점점 정규속도에서 속도가 느려지기 시작하면서 시간과의 관계가 나오고 있다. 그리고 하행선과 다른 점은 속도가 정규속도보다 아주 빠른 경우에 시간과의 관계가 다른 집합에 비해 아주 크다는 것을 알 수 있다.



(KKN10003 : 서초IC, KKN10004 : 양재IC, KKN10002 : 반포IC, KKN10013 : 오산IC, KKN10005 : 판교JC)

속도가 느려지는 시간대는 오후 3시부터 6시대임을 알 수 있습니다.



(KNN10004 : 양재IC)

속도가 아주 빠른 시간대가 새벽임을 알 수 있다.

▪ 가설 2. : '가'지역의 속도가 A가 되면 '나'지역도 속도가 A가 되는가?

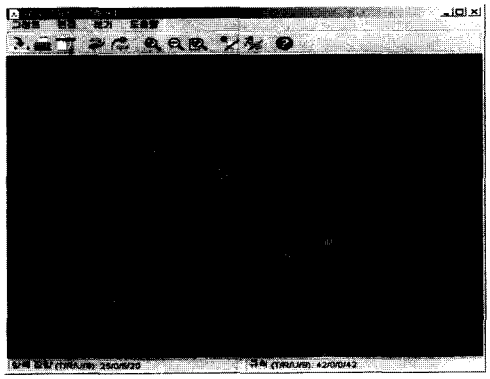
고속도로에서 운전을 하다보면 어느 구간에서 속도가 저속도가 나오면 저속도가 똑같이 나는 구간이 존재 할 것이다라는 생각에서 위 가설을 만들었다. 가설을 검증하는 방법은 아래의 방법으로 실행합니다.

1) 하행선에 대한 구간에 대하여 연관화를 실행
2) 상행선에 대한 구간에 대하여 연관화를 실행

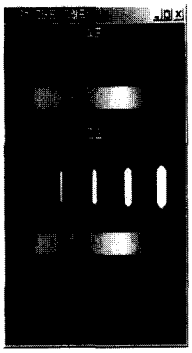
▪ 연관화 1: 하행선에 대하여 연관화를 실행한다

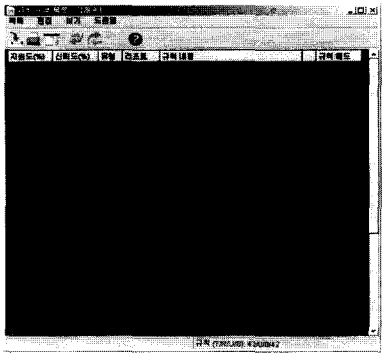
- 방법: 1. 속도가 연속된 값으로 되어 있어서 연관성을 찾는데 많은 트랜잭션이 생기게 되므로 속도에 대해서 이산화 작업을 수행한다.
2. 속도는 0부터 200까지 5차이로 나누어지도록 한다.
3. 추가 자료를 만든 후, 이산화 된 속도에 대하여 구간을 연관화 한다.

▪ 결과 화면



▪ 자료1





- 자료2 자료3

(KKN10006 : 판교IC, KKN10010 : 수원IC, KKN10008 : 죽전휴게소, KKN10007 : 서울TG)

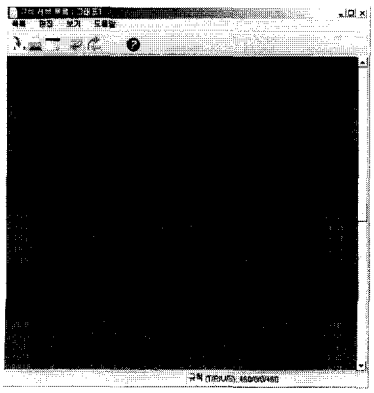
- 결과고찰

  위 자료3에 의하면 판교IC, 수원IC, 죽전휴게소의 속도가 같으면 서울TG도 일정속도가 된다. 수원IC, 죽전휴게소의 속도가 같으면 서울TG도 일정속도가 된다. 판교IC, 수원IC의 속도가 같으면 서울TG도 일정속도가 된다. 수원IC의 속도로 서울TG의 속도를 알 수 있다. 따라서 위 4개의 지역은 서로 많은 연관관계를 가지고 있는 것을 알 수 있다. 서울TG의 속도를 빠르게 하려면 위 3곳의 속도를 빠르게 하면 될 것이라고 본다. 결론적으로 하행선에서는 연관화가 존재한다는 것을 알 수 있다.

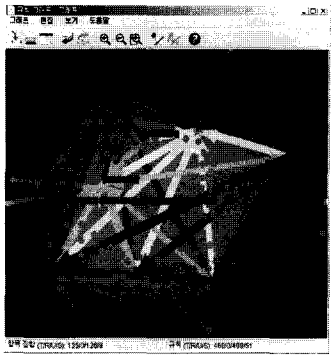
- 연관화 2: 상행선에 대하여 연관화를 실행한다.

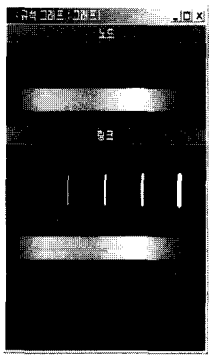
  - 방법: 1. 연관화1의 과정과 같다.

- 결과 화면



- 자료1





- 자료2 자료3

(KKN10011 : 기흥휴게소, KKN10012 : 기흥IC, KKN10013 : 오산IC, KKN10008 : 죽전휴게소)

- 결과 고찰

  위 자료1과 자료2를 보면 기흥휴게소의 속도와 죽전휴게소의 속도가 관계가 있다. 죽전휴게소의 속도와 오산IC의 속도가 관계가 있다. 따라서 상행선에서도 속도에 대하여 연관성이 있는 구간이 있다.

- 가설 3. 톨게이트 인접 지역은 평균 속도가 40km/h 이하이다.

  평균 속도가 40km/h 이하라는 검증을 위하여 우리는 visualization을 통하여 검증하고자 하였다. visualization으로 2가지를 보였다. 하나는 전일, 전 시간대의 평균 속력을 구하는 것이며, 다른 하나는 모든 날의 시간당 평균 속력을 구하는 것이다. 전체 평균 속도를 보이는 것은 텍스트로서 쉽게 이해가 가나, 시간당 평

균 속도는 텍스트를 통한 visualization은 이해의 한계가 있으므로, 그래픽한 visualization을 동반하였다.

- Visualization 1: 서울 톨게이트의 상·하행선 평균 속도를 보여준다.

- 방법: 1. tollgate라는 이름의 테이블을 생성한다.
2. 서울 톨게이트와 인접한 노드를 정보를 삽입한다.
3. 아래와 같은 질의를 통하여 톨게이트로 서울에서 진입하는 도로(KUT10070)와 서울로 진입하는 도로 (KDT10060)의 10일 간의 평균 속도를 추출한다.

SQL> select link_id, avg(speed) from tollgate group by link_id;

- 결과 화면:

LINK_ID	AVG(SPEED)
KDT10060	80.428505
KUT10070	87.649533

- 결과 고찰
생각 외로 톨게이트와 인접한 도로 속도는 높았다. 기존에 톨게이트 주변이 막힐 것이라는 생각은 잘못된 생각임이 증명되었습니다.

- Visualization 2: 서울 톨게이트의 시간 당 상·하행선 평균 속도를 보여준다.

- 방법: 1. 우리가 시험하고 있는 highway 테이블에 존재하는 ttime 컬럼은 시간과 분 정보를 동시에 가지고 있으므로 시간만으로 그룹화 시키기 위해서는 별도의 컬럼이 필요하다. 따라서 아래와 같은 질의문을 통하여 Visualization 1의 tollgate 테이블에 hour라는 컬럼은 추가시킨다.
ALTER TABLE tollgate ADD COLUMN (hour char(2));

2. 새로 추가한 컬럼에 시간만을 포함하는 정보를 삽입하기 위하여 아래와 같은 질의를 사용한다.

UPDATE tollgate SET hour = substr(ttime,1,2);

3. 해당 도로를 시간별로 그룹화 하여서 9일간 각 시간대의 속도를 그룹화 하여 평균을 구한다. 아래와 같은 질의를 사용한다.

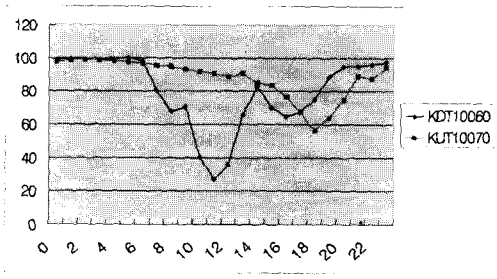
SELECT LINK_ID, HOUR, AVG(speed) FROM TOLLGATE GROUP BY link_id, hour

4. 그래픽한 결과를 보여주기 위하여 엑셀로 포팅(porting)한다.
5. 엑셀로부터 도표를 산출한다.

- 텍스트 결과 화면:

LINK_ID	hour	AVG(SPEED)			
KDT10060	00	98.802083	KUT10070	00	97.302083
KDT10060	01	99.354167	KUT10070	01	98.489583
KDT10060	02	99.020833	KUT10070	02	98.958333
KDT10060	03	98.808511	KUT10070	03	98.797872
KDT10060	04	99	KUT10070	04	98.072289
KDT10060	05	100.2619	KUT10070	05	97.380952
KDT10060	06	97.845238	KUT10070	06	96.630952
KDT10060	07	80.595238	KUT10070	07	95.488095
KDT10060	08	67.46988	KUT10070	08	94.626506
KDT10060	09	70.595238	KUT10070	09	93.083333
KDT10060	10	40.214286	KUT10070	10	91.964286
KDT10060	11	26.845238	KUT10070	11	90.690476
KDT10060	12	35.97619	KUT10070	12	88.595238
KDT10060	13	65.728395	KUT10070	13	90.765432
KDT10060	14	82.464286	KUT10070	14	84.892857
KDT10060	15	70.25	KUT10070	15	83.761905
KDT10060	16	64.776471	KUT10070	16	76.623529
KDT10060	17	67.705263	KUT10070	17	67.526316
KDT10060	18	74.916667	KUT10070	18	56.520833
KDT10060	19	88.715789	KUT10070	19	63.926316
KDT10060	20	94.791667	KUT10070	20	74.739583
KDT10060	21	95.083333	KUT10070	21	88.927083
KDT10060	22	95.677083	KUT10070	22	87.239583
KDT10060	23	97.541667	KUT10070	23	94.145833

▪ 그래픽 결과 화면



(X축:시간, Y축: 속도)

KDT10060 - 서울에서 부산 방향으로, 서울 톨게이트에 진입하는 도로

KUT10070 - 부산에서 서울 방향으로, 서울 톨게이트에 진입하는 도로

▪ 결과 고찰

시간대별 분포를 통해서도 알 수 있듯이 대부분의 시간에서 속력이 40km/h를 초과하였다. 단지 서울 톨게이트로 집입하는 하행선 도로(KDT10060)에서 10시-12시 사이의 속력이 40km/h 이하로 나타났다. 서울로 집입하는 상행선 도로(KUT10070)의 시간 당 평균 속력은 항상 40km/h를 웃돌았다. 하행선이 시간에 대한 속도의 영향이 상행선에 비해 상대적으로 크게 나타나고 있는 것을 볼 수 있다. 특히, 상행선과 하행선에서의 시간당 속력의 그래프는 대부분의 작업이 시작되는 10시에서 12사이에 지체현상을 발생하는 것을 보였으며, 점심을 기점으로 다시 지체 현상을 나타냈으나, 하행선에서 진입하는 도로는 일과가 끝나는 시점에 지체 현상을 보였다. 결과 그래프는 영향받는 정도를 꺽은선 그래프로 나타낸 것이다.

6. 결론

본 논문은 교통 정보 데이터베이스에 데이터 마이닝을 적용해서 고속도로의 속도에 영향을 주는 요소를 도출하고 있으며 정보 데이터베이스 스키마, 데이터 인스턴스, 시스템 구조도를 내포하고 있다. 교통정보 데이터베이스는 도로의 상태, 날씨, 구간, 기상 등의 정보를 포함하고 있으며, 시스템 구조는 이종간의 시스템간의 작업을 처리하기 위해 전처리 과정을 수행할 수 있는 컨버터를 구축하였으며, 컨버터는 데이터 마이닝을 하기 위한 기본 자료를 생성하여 준다. 데이터 마이닝을 수행하기 위해 세 가지 가설을 설정하였으며, 가설에 적합한 마이닝 연산을 적용하여 결과를 도출하였다.

첫 번째 마이닝의 가설은 '속도가 가장 빠른 시간대가 언제인가' 이며, 이를 위해 세 가지 방법(①양방향의 차선에 대한 속도를 클러스터링 한다. ② 한 구간의 자료를 뽑아서 클러스터링 작업으로 시간 분포의 특성을 분석한다. ③ 자료를 나누어서 속도에 대해 클러스터링 한다.)을 클러스터링 하여 속도를 평균속도, 저속도, 과속도로 군집시켜 특성을 도출한 결과 새벽 시간에 제일 빠른 속도가 결과로 제시되었다.

두 번째 마이닝의 가설은 '두 지역간의 속도가 연관이 있는가' 이며, 이를 위해 두 가지 방법(① 하행선에 대한 구간에 대하여 연관화를 수행한다. ②상행선에 대한 구간에 대하여 연관화를 수행한다.)을 적용하였고, 상행선과 하행선 모두 상호간의 구간이 속도에 밀접하게 영향을 주는 지역이 많이 있음을 발견하였다. 마지막 마이닝의 가설은 '톨게이트 인접 지역은 평균 속도가 40km/h 이하이다.' 이며, 이를 위해 두 가지 방법(①전날 전 시간대의 평균 속력을 구하는 방법 ②모든 날의 시간 당 평균 속력을 구하는 방법)에 의하여 톨게이트 인접 지역의 속도는 40km/h이하가 아님을 도출하였다.

세 가지 가설에 의해 도로 교통에 대하 일반적인 편견이 실제 도로 상황에서는 적용이 되지 않음을 알게 되었고, 한 구간이 속도가 느리거나 막히면, 연관되어 있는 다른 구간들이 영향을 받는 일반적인 사실도 확인하게 되었다. 또한 평균 속도에 대해서는 오전, 오후에는 아주 비슷한 속도의 양상을 보였으며, 새벽 시간대에 속도가 빠른 것으로 나타났다. 향후계획으로 다양한 가설에 의해 다양한 연산을 적용하여 좀 더 많은 요소들을 도출하여 제시하는 것이 필요하다.

※ 참고문헌

[1] "IBM DB2 Intelligent Miner for Data", IBM corp., 1999.

[2] I. Witten, E. Frank, "Data Mining", Morgan Kaufmann Publishers, 1999

[3] "Oracle Adminstration Handbook", Oracle press., 198.

[4] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery: An overview. In Advances in Knowledge Discovery and Data Mining", pp. 1-34. AAAI Press, Menlo Park, CA, 1996.

[5] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals", Data Mining and Knowledg, 1997.

[6] M. Chen, J. Han, and P. Yu, "Data mining: An overview from database perspective", IEEE Transactions on Knowledge and Data Eng., 8(6):866--883, December 1996.

[7] M. Holsheimer, M. Kersten, H. Mannila, and H. Toivonen, "A perspective on databases and data mining", In 1st Intl. Conf. Knowledge Discovery and Data Mining, Aug. 1995.

[8] 김정자, 이도헌, "데이터 마이닝 기술 및 연구 동향" 정보과학회지, 16(9):6-14, 9 1998

[9] 오병우, 이강준, 한기준 "공간 데이터 마이닝에 관환 고찰" 정보과학회지, 16(9):45-54, 9 1998.

[10] "Oracle Programmer's Guide", Oracle press., 1998.

이기성

1993년 2월 : 숭실대학교 전자계산학과 졸업
1996년 8월 : 숭실대학교 컴퓨터학과 공학석사
1996년~현재 : 숭실대학교 컴퓨터학과 박사수료
관심분야 : 멀티미디어 통신, 멀티미디어 응용, 무선 이동 통신, 멀티미디어 데이터베이스

박종천

1994년 2월 대전산업대학교 전자계산학(학사)
1998년 2월 숭실대학교 대학원(석사)
2000년 9월~현재 대전기능대학, 멀티미디어과 전임강사
관심분야 : 영상처리, 멀티미디어

김광휘

1971년 2월 경희대학교(학사)
1980년 8월 경희대학교 대학원 (석사)
1981년~현재 우송정보대학 교수