이 학 석 사 학 위 논 문

# 고객 추천 시스템(CRM)에서 선험적 알고리즘과 LASSO의 비교

## A Study on comparative of Apriori Algorithm and LASSO in Customer Relationship Management(CRM)

2014년 8월

서울대학교 대학원

통계학과

김 종 대

# 고객 추천 시스템(CRM)에서 선험적 알고리즘과 LASSO의 비교

## A Study on comparative of Apriori Algorithm and LASSO in Customer Relationship Management(CRM)

지도교수 Myunghee Cho Paik

이 논문을 이학석사학위논문으로 제출함

2014년 4월

서울대학교 대학원

통계학과

김 종 대

김종대의 석사학위논문을 인준함

2014년 6월

위 원 장 김 용 대 (인)

부위원장 Myunghee Cho Paik (인)

위 원 박 태 성 (인)

# 국 문 초 록

## 고객 추천 시스템(CRM)에서 선험적 알고리즘과 LASSO의 비교

## A Study on comparative of Apriori Algorithm and LASSO in Customer Relationship Management(CRM)


유통분야, 마케팅 또는 웹 마이닝 분야에서 상품 추천이나 고객들의 구매패턴의 연관성을 발견하기 위하여 연관규칙분석(association rule analysis) 알고리즘을 사용한다. 연관규칙분석은 대용량 데이터베이스에서 변수들 간의 흥미로운 관계를 찾도록 고안된 방법으로 자료에 존재하는 항목(item)들 간의 if-then 형식의 연관규칙을 찾는 방법으로서 비지도학습의 일종이다.

본 논문에서는 고객 상품 추천 시에 기존의 연관성 분석보다 효과가 좋은 모델을 탐구해보고 기존의 알고리즘과 비교·분석한다. 고객의 구매여부는 Binary형태이므로 로지스틱 회귀 분석을 도입하고, 구체적으로 Zou와 Hastie(2005)에 의해 제안된 elastic net 모델을 비교·평가한다. Elastic net 모델은 능형회귀와 Lasso회귀의 절충으로서, 상관관계가 있는 변수들 중에서 하나의 변수만을 흔히 선택하는 Lasso의 단점을 보완하는 형태이다.

Elastic net모델을 구성하는 능형회귀와 Lasso의 가중치인 $\lambda$와 $\alpha$의 조정과, 기존의 알고리즘보다 효과가 좋은 모델의 지속적인 탐구를 통하여, 향후 CRM의 여러 분야에서 분석 및 예측을 실시하는데 다양한 전략을 구상할 수 있도록 큰 도움이 될 수 있게 한다.



**주요어** : 빅데이터, CRM, 연관성 분석, Elastic net모델, 능형회귀, Lasso, Hit ratio

**학 번** : 2012-23011

# Contents

# List of Tables

# List of Figures

# Chapter 1

# 서론

오늘날 우리는 이른바 빅 데이터(Big Data) 시대에 직면해 있다. 산업분야를 넘어서 사회 전반에 걸쳐 막대한 데이터가 실시간으로 생산 · 축적되는 상황 속에서 이를 잘 활용하여 고부가가치 정보로 전환할 수 있는가가 전 세계적 관심사로 자리잡았다. 따라서 막대한 데이터가 내포하는 의미를 제대로 해석하고 이들의 연관관계를 분석하여 미래를 예측하거나 새로운 재화나 서비스를 창조해 낼 수 있는 연구가 활발히 진행되어야 할 필요성이 그 어느 때 보다 중요한 시점이다.

특히, 유통분야, 마케팅 또는 웹 마이닝 분야에서 상품 추천이나 고객들의 구매패턴의 연관성을 발견하기 위하여 연관규칙분석(association rule analysis) 알고리즘을 사용한다. 연관규칙분석은 대용량 데이터베이스에서 변수들 간의 흥미로운 관계를 찾도록 고안된 방법으로 자료에 존재하는 항목(item)들 간의 if-then 형식의 연관규칙을 찾는 방법으로서 비지도학습의 일종이다. 흔히 기업의 데이터베이스에서 상품의 구매, 서비스 등 일련의 거래 또는 사건들 간의 연관성에 대한 규칙을 발견하기 위해 적용되며, 마케팅에서는 손님의 장바구니에 들어있는 품목 간의 관계를 알아본다는 의미에서 장바구니 분석(market basket analysis)이라고도 부른다.

본 논문의 목적은 고객 상품 추천 시에 기존의 연관성 분석보다 효과가 좋은 모델을 소개하는 데 있다. 자료의 특성 상 고객의 구매여부는 Binary(1 : 구매, 0 : 비구매)형태로 볼 수 있으며 이는 로지스틱 회귀 분석의 사용을 가능케 한다. 즉, 해당 상품을 추천할 때 다른 상품의 구매 여부를 안다고 가정하고, 조건부 확률을 적절한 연결함수(Link function)를 통해 모형화 하는 것이다. 좀 더 구체적으로 Zou와

Hastie(2005)에 의해 제안된 elastic net 모델을 추천할 것이다. elastic net 모델은 능형회귀와 Lasso회귀의 절충으로서, 상관관계가 있는 변수들 중에서 하나의 변수만을 흔히 선택하는 Lasso의 단점을 보완하는 형태이다.

본 논문의 2장에서는 두 가지 방법론의 비교를 위해 사용할 데이터에 대해 소개를 한다. 또한, 방대한 데이터인 만큼 분석 목적에 맞게 데이터 정제를 실시한다. 3장에서는 비교 대상인 두 가지 방법론에 대한 이론적 배경을 소개한다. 특히, elastic net 모델의 경우 로지스틱 회귀, 능형회귀, Lasso 회귀의 기본적인 내용부터 다룬다. 4장에서는 정제된 데이터를 가지고 두 가지 방법론을 hit ratio라는 평가 기준을 바탕으로 비교·분석하며 결과에 대해 토의하고 연구 결과와 비교하여 본 논문이 추천하는 모델의 향상된 점에 대해 고찰한다.

# Chapter 2

# 데이터

## 2.1 데이터 설명

본 논문의 분석에 사용된 데이터는 Groceries data set이다. 이는 Michael Hahsler, Kurt Hornik and Thomas Reutterer에 의해 제공된 데이터로, 통계 프로그램 R의 'arules' 패키지에 내장된 데이터이다. Groceries data set은 실제 일반적인 지역의 식료품 가게에서 1달(30일) 간의 판매기록이다. 총 9835명의 user의 169개 item에 관한 구매 여부를 구매는 1, 비구매는 0을 이용하여 표시한 자료이다. 자료의 특징으로는 9,835×169=1,662,115개의 방대한 자료이며, 그 중 구매기록인 1의 개수는 43367로 전체의 2.61%밖에 안 되는 굉장히 sparse한 자료이다.

| | frankfurter | sausage | liver loaf | ham | meat | finished products | organic sausage | chicken | turkey |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 28 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 2.1 :Groceries data set 일부

## 2.2 데이터 정제

### 2.2.1 구매빈도 합이 17미만인 item제거

각 user별, 각 item별 구매 빈도수의 합계는 다음과 같다.

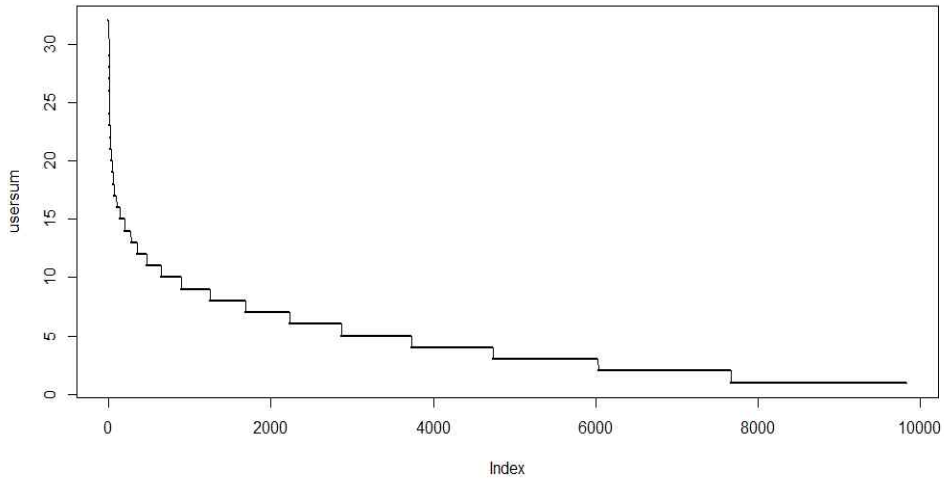


Figure 2.1 : user별 구매빈도수 합계의 plot

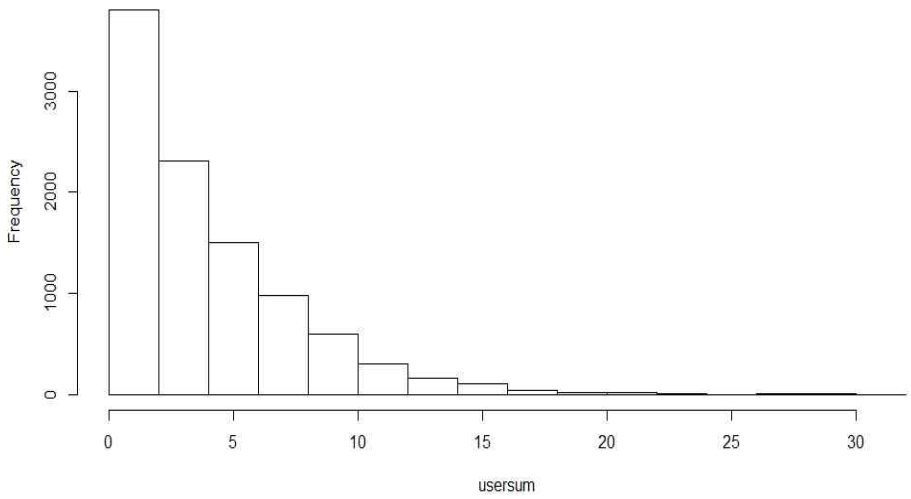


Figure 2.2 : user별 구매빈도수 합계의 히스토그램

| 최소값 | 1분위수 | 중위수 | 평균 | 3분위수 | 최대값 |
|---|---|---|---|---|---|
| 1.000 | 2.000 | 3.000 | 4.409 | 6.000 | 32.000 |

Table 2.2 : user별 구매빈도수 합계의 기초통계량

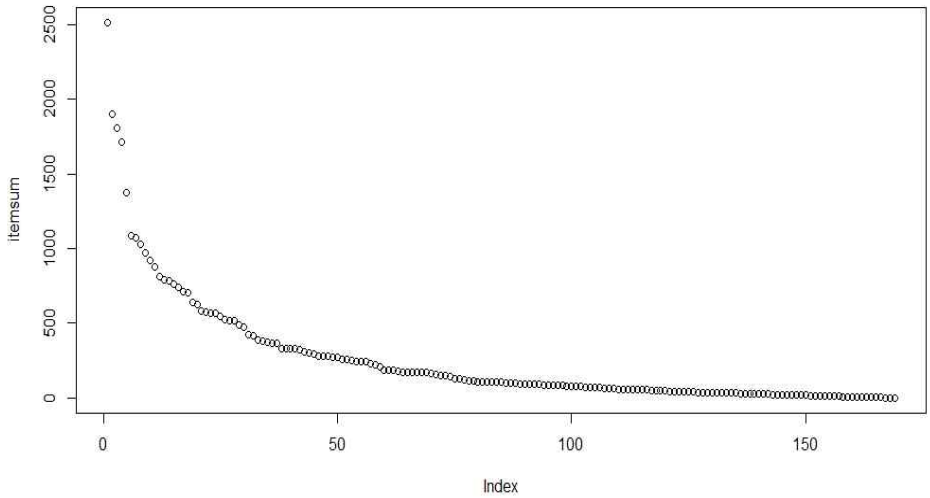


Figure 2.3 : Item별 구매빈도수 합계의 plot

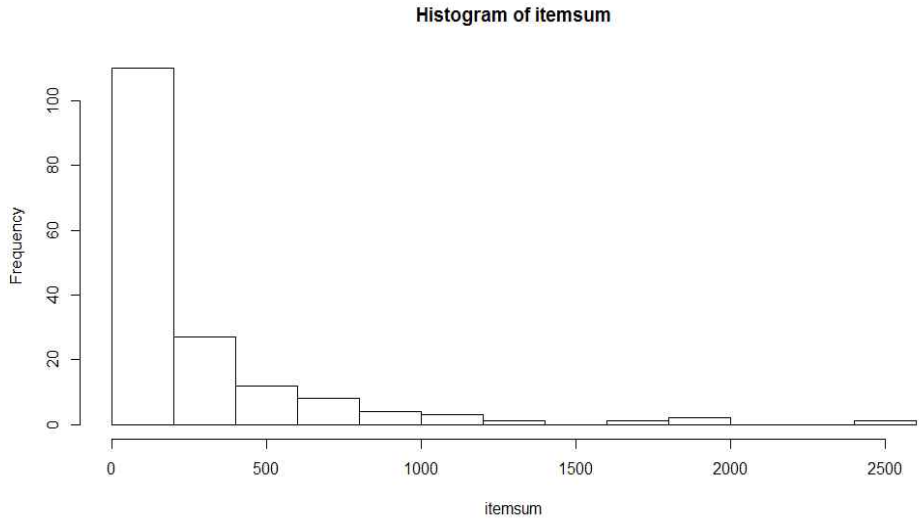


Figure 2.4 : Item별 구매빈도수 합계의 히스토그램

| 최소값 | 1분위수 | 중위수 | 평균 | 3분위수 | 최대값 |
|---|---|---|---|---|---|
| 1.0 | 38.0 | 103.0 | 256.6 | 305.0 | 2513.0 |

Table 2.3 : Item별 구매빈도수 합계의 기초통계량

위의 user별 구매빈도수 합계를 보면 169개 item 중 평균 4.409개의 item을 구매했음을 알 수 있다. 많은 수치는 아니지만 분석에 큰 지장이 없다는 판단 하에 모든 데이터를 사용한다. 그러나 item별 구매빈도의 합의 경우는 다음 두 가지 이유로 정제를 실시했다. 첫 번째로 9835명의 user중 극소수(1~17개)만이 구입한 item은 분석 과정 및 결과 해석에서 큰 의미가 없고 오히려 해석에 방해가 된다는 판단하에 제거하였다. 두 번째로 분석에 사용할 elastic net 모델의 경우 로지스틱 모형의 형태상, 입력변수에 0이 들어가면 회귀계수의 추정이 부정(不正)이 되고, 종속변수에 0이 들어가면 회귀계수의 추정이 불능(不能)이 된다. 즉, item별 구매빈도수 합계가 지나치게 적으면 training set, test set분할 시 어느 한쪽은 전체 벡터가 0이 되는 경우가 발생하여 분석이 불가능하다. 따라서 위의 두 가지 이유로 item별 구매빈도 수의 합계가 17미만인 item을 분석 초기에 제거 한 후 169개 item 중 150개만 분석에 활용하였다.

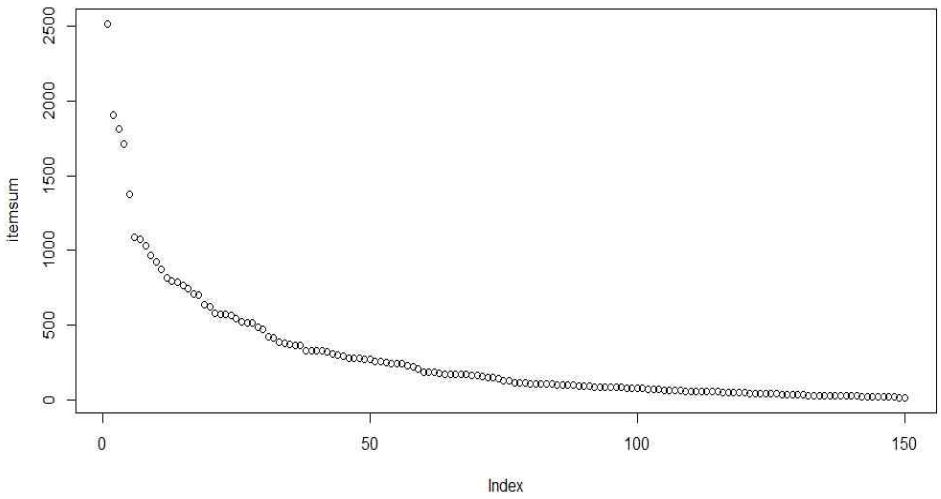


Figure 2.5 : Item제거 후 구매빈도수 합계의 plot

| 최소값 | 1분위수 | 중위수 | 평균 | 3분위수 | 최대값 |
|---|---|---|---|---|---|
| 17.00 | 55.25 | 129.00 | 288.10 | 332.00 | 2513.00 |

Table 2.4 : Item제거 후 구매빈도수 합계의 기초통계량

### 2.2.2 상품명 변경

분석 시 코딩의 편의와 연관성 분석에서 연관규칙을 좀 더 쉽게 도식화 시키고자 상품명을 순서대로 i1 ~ i150으로 변경하였다.

## 2.3 데이터 분할

### 2.3.1 Training set - Test set 분할

분석에 앞서, 9835개의 user 데이터를 7:3의 비율로 training set(6884개)과 test set(2951개)으로 랜덤하게 분할하였다. 즉, 연관성 분석의 경우에는 training set을 이용하여 연관성 규칙을 먼저 생성하고 해당 규칙을 그대로 test set에 적용하여 성능을 평가했다. Elastic net 모델의 경우 training set을 이용하여 회귀계수를 추정하여 모델을 만들고 test set에 그대로 적용하여 성능을 평가했다.

### 2.3.2 Training item - Test item 분할

2.3.1의 방법은 test set에 해당 모델을 적용하여, 한 user의 한 가지 item에 대한 구매가능성의 스코어를 계산할 때, 나머지 item에 대한 구매여부를 모두 안다고 가정한다. 그리고 다시 나머지 item에 대한 구매가능성의 스코어를 계산할 때는 앞서 예측했던 item의 구매여부를 안다고 재 가정한다. 이는 받아들이기에 조금은 불편한 방법이다. 즉, 예측하고자 하는 같은 정보를 한번은 모른다고 가정하고 스코어를 예측하였다가, 나머지의 경우에는 알고 있다고 가정해야 하는 문제가 있다. 그래서 같은 training set-test set분할에 대해서 item분할을 추가로 실시하였다. 총 150개의 item에 대하여 구매빈도수의 합계가 높은 상위 50개와 남은 100개의 item 중 50개를 랜덤으로 뽑아 이들을 합쳐 총 100개의 training item을 선발하고 남은 50개 item에 대해서만 test를 실시한다.

# Chapter 3

# 방법론

## 3.1 연관 규칙 분석

### 3.1.1 연관 규칙

연관 규칙 분석은 자료에 존재하는 항목(item)들 간의 if-then 형식의 연관 규칙을 찾는 방법으로서 비지도학습법의 일종이다. 기업의 데이터베이스에서 상품의 구매, 서비스 등 일련의 거래 또는 사건들 간의 연관성에 대한 규칙을 발견하기 위해 적용된다. 예를 들어 “사이다를 구입하는 고객은 오렌지 쥬스를 산다”와 같이 “If A, then B”와 같은 형식이며 모든 규칙이 유용한 것은 아니다. 유용한 규칙이 되기 위한 필요조건은 두 품목 A와 B를 동시에 구매한 경우의 수가 일정 수준 이상이며 품목 A를 포함하는 거래 중 품목 B를 구입하는 경우의 수도 일정수준 이상이어야 한다. 따라서 판단 기준이 될 몇 가지 척도가 필요하다.

### 3.1.2 연관규칙분석의 척도

ⅰ) 지지도(Support) : 전체 거래 중 항목 A와 항목 B를 동시에 포함하는 거래의 비율로 정의된다. 전체 거래 중 항목 A와 항목 B를 동시에 포함하는 거래가 어느 정도인지를 나타내주며, 전체 구매 경향을 파악 할 수 있다. 그만큼 많이, 같이 판매되고 있다는 뜻으로 연관 규칙이 나왔을 때 적용성이 있는지를 판단할 수 있고 불필요한 분석을 대폭 줄일 수 있다.

$$\text{지지도} = P(A \cap B) = \frac{A\text{와 } B\text{가 동시에 포함된 거래 수}}{\text{전체 거래수}}$$

ii) 신뢰도(Confidence) : 항목 A를 포함한 거래 중에서 항목 A와 항목 B가 같이 포함될 확률은 어느 정도인가를 나타내주며 연관성의 정도를 파악할 수 있다. 이 의미는 조건부 확률로 “A를 구입한 사람 중 B를 구입한 사람의 비율”을 의미하여 이 값이 높아야 한다.

$$\text{신뢰도} = P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{A\text{와 } B\text{를 동시에 포함하는 거래 수}}{A\text{를 포함하는 거래 수}}$$

iii) 향상도(Lift) : A가 주어지지 않았을 때의 품목 B의 확률에 비해 A가 주어졌을 때의 품목 B의 확률의 증가 비율이다. 만일 향상도가 1이면 A의 구매와 B의 구매가 서로 독립인 경우이다. 향상도가 1보다 크면 이 규칙은 결과를 예측하는 데 있어서 우연적이기보다는 우수하다는 것을 뜻한다. 1보다 작으면 이러한 규칙은 우연적 기회보다 도움이 되지 않음을 뜻한다.

$$\text{향상도} = \frac{P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(A)P(B)} = \frac{A\text{와 } B\text{를 포함하는 거래 수}}{A\text{를 포함하는 거래 수} \times B\text{를 포함하는 거래 수}}$$

### 3.1.3 연관규칙분석의 절차

만약 k개의 품목이 있는 경우에 $2^k$ 개의 가능한 품목 집합이 있고, k가 아주 큰 경우에 이 모든 집합 중에 지지도가 높은 집합을 찾는 것은 현실적으로 불가능하다. 이에, 최소 지지도를 갖는 연관규칙을 찾는 대표적인 방법인 Apriori알고리즘이 가장 보편적으로 사용되고 있다. Apriori알고리즘은 다음과 같다.

ⅰ) 최소 지지도를 정한다.

ii) 개별 품목 중에서 최소 지지도를 넘는 모든 품목을 찾는다.

iii) ii)에서 찾은 개별 품목만을 이용해 최소 지지도를 넘는 2가지 품목 집합을 찾는다.

ⅳ) 위의 두 절차에서 찾은 품목 집합을 결합해 최소 지지도를 넘는 3가지 품목 집합을 찾는다.

ⅴ) 반복적으로 수행해 최소 지지도가 넘는 빈발품목 집합을 찾는다.

## 3.2 로지스틱 회귀 분석

로지스틱 회귀는 출력변수 y가 범주형인 경우에 적용할 수 있는 방법이다. 논의의 편의상 $y=0$ 또는 1의 값을 가지고 입력변수가 $x$ 하나인 경우의 모형은 다음과 같다.

$$y = \beta_0 + \beta_1 x + \epsilon$$

위 모형을 적용하는 경우에 $\beta_0 + \beta_1 x$는 범위 [0,1]을 벗어날 수 있고 오차항 $\epsilon$의 분포가 정규분포가 아니라는 문제점이 있다. 이에 대한 대안으로서 연속이고 증가함수이며 [0,1]사이에서 값을 갖는 $p(x)$에 대하여

$$P(Y=1|x) = p(\beta_0 + \beta_1)$$

로 모형화 할 수 있다. 즉, $x$가 주어졌을 때 $Y$의 조건부 평균이 아니라 조건부 확률을 적절한 연결함수(link function) $p$를 통해 모형화 하는 것이 로지스틱 회귀의 아이디어다. 연결함수의 형태에 따라 로지스틱 모형, 검벨(Gumbel), 프로빗(probit) 모형 등이 있는데 본 논문에서는 계산상의 편리성으로 인하여 로지스틱 모형을 사용한다. 단순 로지스틱 모형은 다음과 같다.

$$P(Y=1|x) = \exp(\beta_0 + \beta_1 x)/(1+\exp(\beta_0 + \beta_1 x))$$

이 경우, $\dfrac{P(Y=1|x+1)/P(Y=1|x)}{P(Y=0|x+1)/P(Y=0|x)} = \dfrac{P(Y=1|x+1)P(Y=0|x)}{P(Y=0|x+1)P(Y=1|x)} = \exp(\beta_1)$

을 오즈비(odds ratio)라 한다. 오즈비는 $x$가 한 단위 증가할 때 $y=1$일 확률과 $y=0$일 확률의 비의 증가율을 나타낸다.

단순 로지스틱 모형에 대한 우도함수(likelihood function)는 다음과 같다.

$$L(\beta_0,\ \beta_1) = \prod_{i=1}^{n} p(\beta_0+\beta_1 x_i)^{y_i} (1-p(\beta_0+\beta_1 x_i))^{1-y_i}$$

여기서 $p(x) = \exp(x)/(1+\exp(x))$이다.

로그 우도함수 $l(\beta_0,\ \beta_1)=\sum_{i=1}^{n}(y_i\log p(\beta_0+\beta_1 x_i)+(1-y_i)\log(1-p(\beta_0+\beta_1 x_i)))$ 는 계수에 대한 비선형 함수이므로 이를 최대화하는 최대우도(maximum likelihood) 추정치 $(\widehat{\beta_0},\ \widehat{\beta_1})$은 수치적 방법을 이용하여 구할 수 있다. 입력변수 $x$가 $y$를 설명하는데 유의한지에 대한 유의성 검정은 우도비 검정 통계량 $\chi^2=-2(\max l(\beta_0,\ 0)-l(\widehat{\beta_0},\ \widehat{\beta_1}))$으로 주어진다.

이제 단순 로지스틱 회귀를 다중 및 다항 로지스틱 회귀로 확장할 수 있다. 출력변수가 이항 범주인 경우 다항 로지스틱 회귀모형은 입력변수 $x$에 대하여 $P(Y=1\mid x)=p(\beta_0+\beta_1 x+\cdots+\beta_p x^p)$이며, 다중 로지스틱 회귀모형은 입력변수 $x_1,\ \cdots,x_p$에 대하여 $P(Y=1\mid x)=p(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)$이다.

여기서 $p(x)=\exp(x)/(1+\exp(x))$이고 모수 $\beta_0,\ \cdots\beta_p$는 최대우도법으로 추정될 수 있으며 유의성 검정은 단순 로지스틱 회귀와 마찬가지로 우도비 검정을 이용한다. 즉, 기존의 구매 데이터를 입력변수로 사용하여 예측하고자 하는 출력변수의 구매여부를 확률로 표현하기에 적합한 모형임을 알 수 있다.

## 3.3 벌점화 방법

회귀모형에서 $\beta$의 추정에 쓰이는 방법 중 벌점화 방법에 대해 알아보고, 대표적 방법인 능형회귀와 Lasso회귀에 대해 알아본다. 벌점화 기법의 기본적인 아이디어는 불편성에서 벗어나서 편의는 있지만 분산을 더 작게 만들어 전체 평균제곱오차(MSE) 측면에서 기존의 최소제곱추정치보다 좋은 추정치를 찾는 것이다. $\theta$에 대한 추정치 $\tilde{\theta}$의 평균제곱오차(mean squared error, MSE)는 다음과 같다.

$$MSE(\tilde{\theta})\ =\ E(\tilde{\theta}\ -\ \theta)^2\ =\ Var(\tilde{\theta})+[E(\tilde{\theta})\ -\ \theta]^2$$

즉, 평균제곱오차는 추정치의 분산과 편의의 제곱의 합으로 이루어진다. 가우스 마코프 정리(Gauss Markov Theorem)에 의하면 회귀문제에서 최소제곱 추정치는 선형 불편(unbiased) 추정치 중에서 분산이 최소이다. 불편성에서 벗어나서 생각해보면 편의가 있지만 분산이 작아서 전체 평균제곱오차 측면에서 최소제곱 추정치보다 좋은 추정치가 존재할 수 있다. 이 아디이어에 기반을 둔 것이 소위 축소추정

(shrinkage estimation)이다. 다음에 소개되는 능형회귀(ridge regression) 및 lasso회귀(least absolute shrinkage and selection operator)는 이러한 축소 추정법을 이용한 방법이다.

### 3.3.1 능형회귀(Ridge regression)

능형회귀 추정치는 제약조건(constraint) $\sum_{j=1}^{p} \beta_j{}^2 \leq t^2$ 하에서

$$\hat{\beta}^{ridge} = \text{argmin}_{\beta} \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2$$

로 주어지며 라그랑지 승수법(Lagrange multiplier)에 의하면 다음과 동치이다.

$$\hat{\beta}^{ridge} = \text{argmin}_{\beta} \left\{ \sum_{i=1}^{n} \left( y_i - \beta_0 \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j{}^2 \right\}$$

여기서 $t \geq 0$이고, $\lambda \geq 0$이다. $t = 0$이면 모형은 상수항만을 포함하고 반대로 $t = \infty$이면 최소제곱법과 동일하다. 이 때 조율모수 $t$ 또는 $\lambda$의 선택은 교차확인법을 흔히 사용한다.

본래 능형회귀는 $p > n$일 때 최소제곱 추정치를 계산하기 위해 고안되었다. $x_{ij}$를 $x_{ij} - \overline{x_j}$로 그리고 $y_i$를 $y_i - \overline{y_i}$로 대체하면 상수항이 없는 회귀모형을 고려할 수 있다. $X$를 $n \times p$ 행렬이라 하면 통상적인 최소제곱 추정치는 $\hat{\beta}^{LS} = (X^T X)^{-1} X^T y$로 주어지나 $p > n$인 경우에는 $(X^T X)^{-1}$가 존재하지 않는다. 따라서 정규방정식 $(X^T X)\hat{\beta} = X^T y$의 해는 유일하지 않다. 능형회귀 추정치는 위의 정규방정식에서 역행렬이 존재하도록 다음과 같이 $X^T X$에 대각행렬 $\lambda I$를 더한 것이다.

$$\hat{\beta}^{ridge} = (X^T X + \lambda I)^{-1} X^T y$$

입력변수들이 서로 직교하는 경우(즉, $X^T X = I$ )에는 $\hat{\beta}^{ridge} = \hat{\beta}^{LS} / (1 + \lambda)$이다. $\lambda \geq 0$이므로 능형회귀 추정치 $\hat{\beta}^{ridge}$는 최소제곱 추정치 $\hat{\beta}^{LS}$의 축소된 추정치로 볼 수 있다. 이 추정치는 편의는 있지만 분산을 줄이면서 전반적인 예측오차를 줄이는데 효과적이다.

### 3.3.2 Lasso 회귀 분석

능형회귀는 축소 추정치를 주지만 변수선택을 하지 않는다. 따라서 좀 더 고차원 자료의 경우에는 Tibshirani(1996)가 제안한 lasso 회귀가 축소와 변수선택을 통해 예측력을 향상시키는 동시에 최종 모형에 대한 해석을 용이하게 한다는 점에서 장점이 있다. lasso 추정치는 제약조건 $\sum_{j=1}^{p}|\beta_j| \le t$ 하에서

$$\hat{\beta}^{lasso} = \text{argmin}_{\beta}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}x_{ij}\beta_j\right)^2$$

이며 능형회귀와 마찬가지로 라그랑지 승수법에 의하면 다음과 동치이다.

$$\hat{\beta}^{lasso} = \text{argmin}_{\beta}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta|\right\}$$

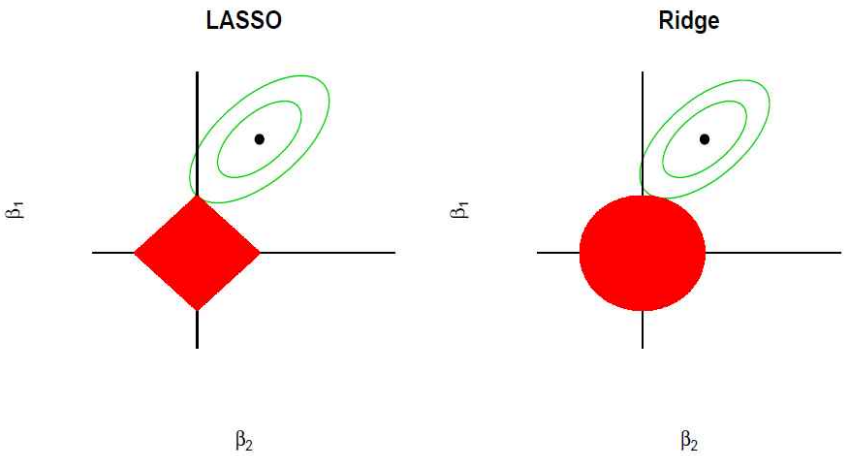


**Figure 3.1 : 능형회귀와 lasso 회귀의 비교**

능형회귀와 lasso 회귀의 차이는 벌점이 $l_2$-노음(norm) $\sum_{j=1}^{p}\beta_j^{\,2}$에서 $l_1$-노음(norm)인 $\sum_{j=1}^{p}|\beta_j|$로 바뀌었다는 점이다. 그림 3.1은 변수가 2개인 경우에 대하여 이 둘의 차이를 보여준다. 최소제곱추정치 $\hat{\beta}^{LS}$를 중심으로 그려진 타원들은 오차제곱합의 등고선들을 보여준다. 안이 채워진 부분은 제약조건 $|\beta_1|+|\beta_2| \le t$와 $\beta_1^{\,2} + \beta_2^{\,2} \le t^2$을 만족시키는 계수값의 영역을 나타낸다. 각 계수 추정치는 등고선과 제약조건 영

역이 만나는 점으로 주어지게 된다. 능형회귀의 경우 제약조건 영역이 원이므로 추정값이 모서리에 닿을 확률이 작다. 이에 비해 lasso는 영역이 정사각형이므로 상대적으로 추정값이 모서리에 닿아 계수값이 0이 될 가능성이 많다. lasso는 설명력이 없는 입력변수들의 계수를 0으로 추정함으로써 자동적인 변수선택이 이루어지는 것이다. 또한 최적 부분집합 선택법은 특정 입력변수를 선택하거나 배제하는 불연속적인 변수선택법인 반면, lasso는 축소를 통해 연속적으로 변수를 선택한다는 장점을 지닌다.

### 3.3.3 Elastic net 모델

Zou와 Hastie(2005)에 의해 제안된 elastic net 벌점은 $l_1$-노음과 $l_2$-노음의 블록결합(convex combination)형태로 다음과 같이 주어진다.

$$\lambda\sum_{j=1}^{p}((1-\alpha)|\beta_j|+\alpha{\beta_j}^2),\alpha\in[0,1]$$

즉, elastic net은 능형회귀와 lasso회귀의 절충으로서, 상관관계가 있는 변수들 중에서 하나의 변수만을 흔히 선택하는 lasso의 단점을 보완하기 위해 제안되었다. Elastic net은 $l_1$-노음에 $l_2$-노음을 결합함으로써 상관관계가 있는 변수들을 모두 선택하도록 하며, 이를 그룹화 효과(grouping effect)라고 부른다.

Elastic net의 특징을 보다 자세히 살펴보면, 입력변수가 두 개인 회귀분석의 경우 $\alpha$와 $\lambda$값들이 주어졌을 때 두 변수의 추정계수를 각각 $\hat{\beta}_1(\alpha,\lambda)$와 $\hat{\beta}_2(\alpha,\lambda)$라 하자. 그러면 어떤 상수 $M>0$에 대하여 다음  부등식이 성립한다.

$$|\hat{\beta}_1(\alpha,\lambda)-\hat{\beta}_2(\alpha,\lambda)|<\frac{\sqrt{n}M}{\alpha\lambda}\sqrt{2(1-r_{12})}$$

여기서 $r_{12}$는 두 변수의 상관계수를 나타낸다. 따라서 두 변수의 상관계수가 1에 가까울수록 두 계수의 차이는 0이 된다. 즉, 두 변수들간의 상관관계가 크면 대응되는 추정치들의 값은 거의 동일하다.

# Chapter 4

# 분석 및 결과

## 4.1 Apriori 알고리즘

### 4.1.1 Item 분할 전

생성된 training set을 이용하여 apriori 알고리즘을 실행한다. 이 때, 데이터의 특성을 고려하여, 지지도가 0.01이상, 신뢰도가 0.1이상인 연관규칙만 분석에 사용한다. 선별된 연관규칙은 다음과 같다.

| | rules | support | confidence | lift |
|---|---|---|---|---|
| 1 | {} => {i95} | 0.11011040 | 0.1101104 | 1.0000000 |
| 2 | {} => {i15} | 0.10488088 | 0.1048809 | 1.0000000 |
| 3 | {} => {i20} | 0.10778617 | 0.1077862 | 1.0000000 |
| 4 | {} => {i30} | 0.13654852 | 0.1365485 | 1.0000000 |
| 5 | {} => {i96} | 0.17431726 | 0.1743173 | 1.0000000 |
| 6 | {} => {i52} | 0.17954678 | 0.1795468 | 1.0000000 |
| 7 | {} => {i23} | 0.19538059 | 0.1953806 | 1.0000000 |
| 8 | {} => {i25} | 0.25159791 | 0.2515979 | 1.0000000 |
| 9 | {i78} => {i25} | 0.01002324 | 0.5847458 | 2.3241281 |
| 10 | {i5} => {i23} | 0.01031377 | 0.3922652 | 2.0076978 |
| 11 | {i5} => {i25} | 0.01045904 | 0.3977901 | 1.5810547 |
| 12 | {i37} => {i25} | 0.01060430 | 0.4534161 | 1.8021459 |
| 13 | {i123} => {i96} | 0.01031377 | 0.3333333 | 1.9122222 |
| 14 | {i36} => {i25} | 0.01060430 | 0.4319527 | 1.7168373 |
| 15 | {i29} => {i23} | 0.01234747 | 0.4086538 | 2.0915785 |
| 16 | {i29} => {i25} | 0.01220221 | 0.4038462 | 1.6051252 |
| 17 | {i65} => {i25} | 0.01045904 | 0.3850267 | 1.5303257 |
| 18 | {i136} => {i25} | 0.01191168 | 0.3849765 | 1.5301261 |
| 19 | {i21} => {i23} | 0.01452644 | 0.4807692 | 2.4606806 |
| 20 | {i21} => {i25} | 0.01147589 | 0.3798077 | 1.5095821 |
| 21 | {i18} => {i30} | 0.01016851 | 0.3043478 | 2.2288622 |
| 22 | {i18} => {i23} | 0.01045904 | 0.3130435 | 1.6022240 |
| 23 | {i18} => {i25} | 0.01147589 | 0.3434783 | 1.3651873 |
| 24 | {i12} => {i23} | 0.01365485 | 0.4104803 | 2.1009269 |
| 25 | {i12} => {i25} | 0.01438117 | 0.4323144 | 1.7182751 |
| 26 | {i110} => {i96} | 0.01031377 | 0.2709924 | 1.5545929 |
| 27 | {i110} => {i23} | 0.01104009 | 0.2900763 | 1.4846732 |
| 28 | {i110} => {i25} | 0.01074956 | 0.2824427 | 1.1225958 |
| 29 | {i115} => {i96} | 0.01016851 | 0.2723735 | 1.5625162 |
| 30 | {i115} => {i25} | 0.01176641 | 0.3151751 | 1.2526936 |
| 31 | {i68} => {i23} | 0.01162115 | 0.3418803 | 1.7498173 |
| 32 | {i68} => {i25} | 0.01641488 | 0.4829060 | 1.9193561 |
| 33 | {i114} => {i30} | 0.01016851 | 0.2755906 | 2.0182610 |
| 34 | {i114} => {i23} | 0.01133062 | 0.3070866 | 1.5717355 |
| 35 | {i114} => {i25} | 0.01438117 | 0.3897638 | 1.5491535 |
| 36 | {i28} => {i30} | 0.01002324 | 0.2653846 | 1.9435188 |
| 37 | {i28} => {i96} | 0.01104009 | 0.2923077 | 1.6768718 |
| 38 | {i28} => {i23} | 0.01191168 | 0.3153846 | 1.6142065 |
| 39 | {i28} => {i25} | 0.01423591 | 0.3769231 | 1.4981169 |
| 40 | {i101} => {i150} | 0.01220221 | 0.1535649 | 1.5569083 |
| 41 | {i150} => {i101} | 0.01220221 | 0.1237113 | 1.5569083 |
| 42 | {i101} => {i96} | 0.01452644 | 0.1828154 | 1.0487508 |
| 43 | {i101} => {i52} | 0.01205694 | 0.1517367 | 0.8451098 |
| 44 | {i38} => {i30} | 0.01162115 | 0.3018868 | 2.2108390 |
| 45 | {i38} => {i23} | 0.01350959 | 0.3509434 | 1.7962040 |
| 46 | {i38} => {i25} | 0.01641488 | 0.4264151 | 1.6948277 |
| 47 | {i8} => {i20} | 0.01074956 | 0.2560554 | 2.3755864 |
| 48 | {i8} => {i23} | 0.01743173 | 0.4152249 | 2.1252106 |
| 49 | {i8} => {i25} | 0.01685067 | 0.4013841 | 1.5953395 |
| 50 | {i53} => {i30} | 0.01016851 | 0.2364865 | 1.7318861 |
| 51 | {i53} => {i96} | 0.01031377 | 0.2398649 | 1.3760248 |
| 52 | {i53} => {i23} | 0.01467170 | 0.3412162 | 1.7464182 |
| 53 | {i53} => {i25} | 0.01772225 | 0.4121622 | 1.6381780 |
| 54 | {i118} => {i96} | 0.01321906 | 0.2774390 | 1.5915752 |
| 55 | {i118} => {i52} | 0.01176641 | 0.2469512 | 1.3754144 |

**Figure 4.1 Training set에서 선별된 연관규칙의 일부**

첫 번째 시뮬레이션에서 선별된 연관규칙은 총 446개이며, 이를 이용해 test set에 적용하여 예측을 실시한다. 각 user별로 해당 item을 구입할 가능성을 score를 통해 비교를 실시한다. Score는 다음과 같이 계산한다. item1의 score를 계산하는 경우, 나머지 item의 구매여부를 모두 안다고 가정하고, 위에서 선별된 연관규칙에

적용되는 if-then규칙이 존재하면 해당 규칙의 신뢰도를 score로 이용한다. 즉, item1을 제외한 나머지 item의 구매여부를 이용해서, 구매한 item 기록이 선별된 연관규칙의 “if”에 있고, “then”부분에 item1이 있는 규칙을 골라낸다. 이 때, 해당하는 규칙이 존재하지 않는 경우, score는 0이 되고, 해당하는 규칙이 여러 개인 경우에는 각 신뢰도의 max값이 score가 된다. 이와 같은 방법으로 item150까지 반복을 하면 각 user별 item의 score가 모두 계산이 되고, 이 중 score가 가장 높은 10개의 item을 각 user에게 추천하는 방법이다.

### 4.1.2 Item 분할 후

같은 apriori알고리즘을 평가하는 데 있어서 이번에는 test set의 item또한 분할하여 실시하였다. Item 분할 전의 방법은, user의 item에 대한 구매가능성의 score를 계산할 때, 나머지 item에 대한 구매여부를 모두 안다고 가정한다. 그러나 다른 item에 대한 score를 계산할 때는 앞서 예측했던 item의 구매여부를 안다고 재 가정하므로 받아들이기에 조금 불편할 수도 있다. 또한, 구매빈도 수의 합이 높은 item을 선별하여 다른 item의 상품추천을 실시하는 것이 실제 활용에 있어서도 많은 도움이 될 것이란 판단 하에 item분할을 실시하였다. 이 경우에도 예측에 사용되는 연관규칙은 4.1.1에서의 연관규칙과 동일하다. 다만, test set에서 item의 score 예측시, user별로 training item을 동일하게 이용하여 test item의 score만 계산하고, 개수가 50개로 앞의 방법보다 적으므로, score가 높은 test item 중 5개의 상품을 user에게 추천하는 방법이다.

## 4.2 Elastic net 모델

### 4.2.1 Item 분할 전

Elastic net 모델의 경우 glmnet패키지를 이용하여 binomial설정 후, 능형회귀와 Lasso회귀의 가중치인 $\lambda$와 $\alpha$를 각각 0.2와 0.01로 설정하였다. 추천 상품의 기준이 될 score는 로지스틱 회귀를 이용하여 다른 item의 구매여부가 주어졌을 때 해당 item을 구입할 조건부확률을 그대로 사용하였다. 즉, training set을 이용하여 item1부터 item150까지 자기 자신을 제외한 나머지 item들로 회귀모형에 적합하여 150개의 회귀계수를 추정한 후, 다시 test set에서 나머지 item이 주어졌을 때 해당 item을 구입할 확률을 각 user별로 직접 계산하였다. Apriori 알고리즘과 마찬가지로 각

user별로 score가 높은 10개의 상품을 추천하였다.

| ITEM | 추정된 회귀계수 | ITEM | 추정된 회귀계수 |
|---|---|---|---|
| (Intercept) | -3.03495 | i76 | . |
| i2 | 0.165918 | i77 | -0.07708 |
| i3 | . | i78 | . |
| i4 | 0.097923 | i79 | -0.10374 |
| i5 | 0.233523 | i80 | . |
| i6 | . | i81 | 0.156348 |
| i7 | 0.063958 | i82 | 0.202915 |
| i8 | 0.064822 | i83 | 0.013538 |
| i9 | 0.039871 | i84 | . |
| i10 | 0.141015 | i85 | 0.018468 |
| i11 | 0.010251 | i86 | . |
| i12 | 0.107119 | i87 | 0.150338 |
| i13 | . | i88 | . |
| i14 | 0.027344 | i89 | -0.07522 |
| i15 | 0.061898 | i90 | -0.04238 |
| i16 | 0.029957 | i91 | 0.036213 |
| i17 | . | i92 | 0.005586 |
| i18 | . | i93 | 0.123567 |
| i19 | . | i94 | 0.06747 |
| i20 | 0.08153 | i95 | 0.005375 |
| i21 | 0.139447 | i96 | 0.008584 |
| i22 | . | i97 | -0.00845 |
| i23 | 0.053125 | i98 | . |
| i24 | 0.107759 | i99 | . |
| i25 | 0.034085 | i100 | . |
| i26 | 0.049375 | i101 | -0.06318 |
| i27 | 0.004354 | i102 | . |
| i28 | 0.081971 | i103 | . |
| i29 | 0.084606 | i104 | . |
| i30 | 0.008413 | i105 | -0.088 |
| i31 | 0.013957 | i106 | -0.04756 |
| i32 | -0.03764 | i107 | -0.0891 |
| i33 | 0.027697 | i108 | . |
| i34 | . | i109 | . |
| i35 | 0.121376 | i110 | 0.067735 |
| i36 | 0.036069 | i111 | . |
| i37 | . | i112 | 0.100274 |
| i38 | . | i113 | -0.09228 |
| i39 | 0.116143 | i114 | -0.00295 |
| i40 | 0.019864 | i115 | . |
| i41 | 0.083572 | i116 | 0.006834 |
| i42 | . | i117 | 0.12022 |
| i43 | 0.2381 | i118 | 0.005015 |
| i44 | . | i119 | -0.13311 |
| i45 | 0.048214 | i120 | . |

| ITEM | 추정된 회귀계수 | ITEM | 추정된 회귀계수 |
|---|---|---|---|
| i46 | 0.007709 | i121 | . |
| i47 | 0.182788 | i122 | 0.102234 |
| i48 | . | i123 | . |
| i49 | 0.126925 | i124 | 0.148463 |
| i50 | 0.25362 | i125 | 0.148303 |
| i51 | 0.085536 | i126 | -0.1518 |
| i52 | 0.170829 | i127 | 0.023161 |
| i53 | 0.142718 | i128 | -0.16911 |
| i54 | 0.120306 | i129 | . |
| i55 | 0.066452 | i130 | . |
| i56 | 0.027538 | i131 | . |
| i57 | 0.11685 | i132 | 0.085605 |
| i58 | 0.214353 | i133 | . |
| i59 | 0.807678 | i134 | . |
| i60 | 0.106491 | i135 | 0.084816 |
| i61 | -0.02075 | i136 | . |
| i62 | . | i137 | -0.00325 |
| i63 | 0.030186 | i138 | . |
| i64 | . | i139 | . |
| i65 | 0.098037 | i140 | . |
| i66 | 0.239342 | i141 | 0.060097 |
| i67 | 0.256376 | i142 | . |
| i68 | 0.113569 | i143 | . |
| i69 | 0.452772 | i144 | . |
| i70 | 0.476097 | i145 | . |
| i71 | 0.236905 | i146 | . |
| i72 | . | i147 | -0.04807 |
| i73 | 0.321404 | i148 | -0.01022 |
| i74 | 0.099135 | i149 | . |
| i75 | . | i150 | 0.050571 |

Table 4.1 : Item1을 나머지 item의 구매여부로 적합한 후 추정된 회귀계수

### 4.1.2 Item 분할 후

Apriori 알고리즘과 마찬가지로, elastic net 모델에서도 item분할을 실시하였다. 이때 같은 시뮬레이션 내에서 분할된 training item과 test item의 종류는 apriori알고리즘과 동일하다. 그러나, apriori알고리즘에서는 training set전부를 이용해서 연관규칙을 선별했던 것과는 달리, elastic net 모델에서는 회귀계수를 추정할 때에도 training set내에서 training item만을 설명변수로 사용하였다. 즉, 선별된 test item들을 가지고 50개의 회귀계수를 추정하여 이를 test data내의 test item들에 관해서만 score를 계산하였다. Apriori알고리즘과 마찬가지로 추천 item이 많지 않으므로 socre가 높은 5개의 item을 선별하여 각 user별로 추천하였다.

## 4.3 결과

Apriori 알고리즘과 Elastic net 모델의 성능을 item 분할 전과 분할 후로 각각 비교를 하였다. 이 때 평가기준은 Hit ratio이다. Hit ratio는 item 분할 전 추천한 10개의 상품과 item 분할 후 추천한 5개의 상품이 실제 데이터에서 구매했는지를 비교하여 그 개수를 세는 것이다. 즉, 이미 item의 구매여부는 알려져 있기 때문에, 해당 모형에서 계산된 score에 기반을 둔 상품 추천이 실제 구매기록과 같은 결과를 냈는지를 비교하면 두 모형의 예측력을 비교할 수 있다. Score의 계산은 training set과 test set, 또한 item 분할에 따라 달라지므로, 총 20번의 시뮬레이션을 실시하였고 각각의 hit ratio의 평균은 다음과 같다.

| 시뮬레이션 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Apriori | 1.337513 | 1.340224 | 1.331074 | 1.300915 | 1.35039 |
| Elastic | 1.525585 | 1.535412 | 1.498475 | 1.50593 | 1.551 |
| 시뮬레이션 | 6 | 7 | 8 | 9 | 10 |
| Apriori | 1.298882 | 1.33243 | 1.316842 | 1.306676 | 1.322264 |
| Elastic net | 1.509658 | 1.5449 | 1.513385 | 1.531345 | 1.510335 |
| 시뮬레이션 | 11 | 12 | 13 | 14 | 15 |
| Apriori | 1.317181 | 1.272789 | 1.330735 | 1.324297 | 1.328363 |
| Elastic net | 1.499831 | 1.463233 | 1.545239 | 1.528973 | 1.548289 |
| 시뮬레이션 | 16 | 17 | 18 | 19 | 20 |
| Apriori | 1.307015 | 1.333446 | 1.269739 | 1.28736 | 1.318536 |
| Elastic net | 1.504914 | 1.545578 | 1.501525 | 1.501186 | 1.52084 |

Table 4.2 : Item 분할 전 20번의 시뮬레이션에 대한 Hit ratio 평균 비교

| 시뮬레이션 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Apriori | 0.048797 | 0.059302 | 0.062352 | 0.085734 | 0.04202 |
| Elastic | 0.146052 | 0.130803 | 0.115215 | 0.124026 | 0.121315 |
| 시뮬레이션 | 6 | 7 | 8 | 9 | 10 |
| Apriori | 0.089122 | 0.069468 | 0.096239 | 0.070823 | 0.061674 |
| Elastic net | 0.132836 | 0.133514 | 0.139275 | 0.131481 | 0.107421 |
| 시뮬레이션 | 11 | 12 | 13 | 14 | 15 |
| Apriori | 0.065402 | 0.063368 | 0.072857 | 0.071162 | 0.021688 |
| Elastic net | 0.123348 | 0.117248 | 0.133514 | 0.118943 | 0.113182 |
| 시뮬레이션 | 16 | 17 | 18 | 19 | 20 |
| Apriori | 0.090478 | 0.039309 | 0.0837 | 0.061674 | 0.049814 |
| Elastic net | 0.132836 | 0.133514 | 0.139275 | 0.131481 | 0.107421 |

Table 4.3 : Item 분할 후 20번의 시뮬레이션에 대한 Hit ratio 평균 비교

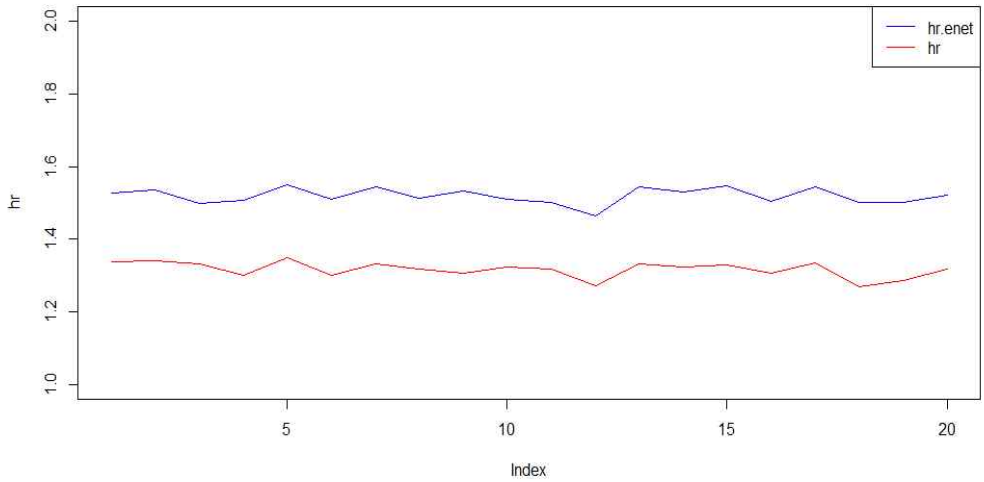


Figure 4.2 : Item 분할 전 20번의 시뮬레이션에 대한 Hit ratio 평균 비교

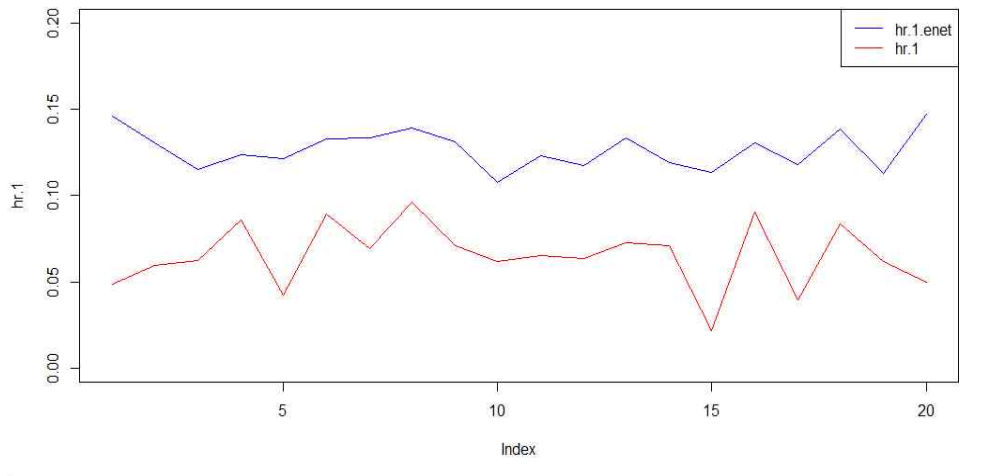


Figure 4.3 : Item 분할 후 20번의 시뮬레이션에 대한 Hit ratio 평균 비교

분석결과, 20번의 시뮬레이션에서 item분할 여부에 관계없이 모두 elastic net모델의 Hit ratio 평균이 높게 나왔다. 즉, 데이터 분할의 여부와 item분할의 여부와 상관없이 구매기록에 기반을 둔 새로운 상품의 추천 시에 기존의 연관성 분석보다 능형회귀와 Lasso에 기반을 둔 회귀분석을 이용한 예측이 더 효과적임을 결론내릴 수 있다.

# Chapter 5

## 맺음말

본 논문에서는 고객 관계 관리(CRM)에서 기존의 선험적 알고리즘인 연관성 분석보다 효과가 좋은 모델을 찾아보고 비교를 하였다. 구체적으로 고객의 구매기록을 통한 새로운 상품 추천 시에 구매여부를 binary로 볼 수 있다고 판단하고 로지스틱 모형의 도입을 생각하였고, Zou와 Hastie(2005)에 의해 제안된 elastic net 모델을 시도해 보았다. Elastic net 모델은 능형회귀와 Lasso회귀의 절충으로서, 상관관계가 있는 변수들 중에서 하나의 변수만을 흔히 선택하는 Lasso의 단점을 보완하는 형태이며, 본 논문에서 사용된 데이터의 경우 데이터가 매우 sparse하여 Lasso의 가중치인 $\alpha$를 0.01로 설정하여 최소화 하였다. 총 20번의 시뮬레이션을 통한 각기 다른 데이터분할과 item분할에서 모두 기존의 알고리즘보다 elastic net모델이 효과가 좋은 것을 확인하였으며, 이는 향후 CRM의 여러 분야에서 데이터분석 및 예측을 실시하는데 효과적인 전략을 구상하는데 큰 도움이 될 것이라 기대한다. 특히, 업계와 데이터의 특성에 따라 가중치인 $\lambda$와 $\alpha$를 조정하면서 다양한 모델을 시도하는 등 효과적인 모델이 지속적으로 개발되길 기대한다.

# Bibliography

박창이·김용대·김진석·송종우·최호식. (2011) R을 이용한 데이터마이닝. 교우사.

Hui Zou and Trevor Hastie. (2005), Regularization and varibale selection via the elastic net. J.R.Statist.Soc.B 67, part2, pp.301-320.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R, (2004) Least angle regression, Ann. statist., 32, 407-499.

Hastie, T., Tibshirani, R. and Friedman, J. (2001) The Elements of Statistical Learning; Data Mining, Inference and Prediction. New York: Springer

Tibshirani, R.(1996) Regression shrinkage and selection via the lasso. J.R.Statist. Soc. B, 58, 287-288

Asela Gunawardana and Christopher Meek. (2008) Tied Boltzmann Machines for cold start recommendations, RecSys'08, October 23-25.

서강수. (2014) 데이터분석 전문가 가이드. 한국데이터베이스진흥원.

# Abstract

Jong Dae Kim
The Department of Statistics
The Graduate School
Seoul National University

Association rule analysis algorithm is extensively employed in distribution field, marketing, or web mining areas in order to discover the association of the product recommendation or customer purchasing pattern.

Association rule analysis is designed to investigate intriguing relationship between types of variables in mass database. This association rule analysis is considered one of the types of unsupervised learning to disclose the association rule of "if-then" pattern among items in research materials.

The present study investigates more effective model in customer product recommendation than association rule analysis on previous research and identify its comparative analysis.

In the present study, logistic regression analysis is employed in that the customer purchase is a type of binary and elastic net model designed by Zou and Hastie(2005) is compared and analyzed.

Elastic net is considered a type of compromising model between Ridge regression and Lasso regression. This model supplements the weaknesses of Lasso regression which typically selects the only variable among correlated variables.

From these empirical findings, it is expected to adjust between elastic net model comprising of Ridge regression and Lasso regression, to discover more effective model than previous algorithm, and to improve diverse strategies in analyzing or predicting future studies on Customer Relationship Management (CRM).

**Keywords** : Big data, Association rule analysis, Elastic net model, Ridge regression, Lasso, Hit ratio

**Student Number** : 2012-23011

## R code

```
install.packages("arules")
library(arules)
install.packages("stringr")
library(stringr)
install.packages("glmnet")
library(glmnet)

data(Groceries)
Groceries
temp = as(Groceries,"matrix")

####plot####
usersum=vector()
for (i in 1:9835){
  usersum[i]=sum(temp[i,])
}

usersum=sort(usersum, decreasing=T)
plot(usersum)
hist(usersum)

itemsum=vector()
for (i in 1:169){
  itemsum[i]=sum(temp[,i])
}

itemsum=sort(itemsum, decreasing=T)
plot(itemsum)
hist(itemsum)
summary(usersum)
summary(itemsum)
```

```
####item 제거(17개 이상만 남김)####
ii <- which((apply(temp, 2, sum) >= 17) == "TRUE")
temp <- temp[,ii]

itemsum=vector()
for (i in 1:150){
  itemsum[i]=sum(temp[,i])
}

itemsum=sort(itemsum, decreasing=T)
plot(itemsum)
hist(itemsum)
summary(itemsum)

####변수명 바꾸기####
colnames(temp)=paste('i',1:150,sep='')
colnames(temp)

####function####
scorevec <- function(indi,temp.ts)
{
  score <- rep(0,ncol(temp.ts))
  rule.ext <- names(indi)[indi!=0]
  nrule<-length(rule.ext)
  qq <- comb(rule.ext)
  for(i in 1:ncol(temp.ts))
  {
    if(sum(rslt[,2] %in% colnames(temp.ts)[i]) > 0)
    {
      rslt.ex <- rslt[rslt[,2] %in% colnames(temp.ts)[i],]
      q<-vector()
      for(j in 1:length(qq))
      {
```

```
            w <- which(match(as.cha(rslt.ex[1]), qq[j]) != "NA")
            q <- c(q,w)
          }
          rslt.ex1 <- rslt.ex[q,]
          if(length(q) > 0) score[i] <- max(rslt.ex1[,4])
        }
    }
    score
}

comb <- function(rule.ex)
{
    ky <- rule.ex
    for(i in 1:(length(rule.ex)-1))
      for(j in (i+1):length(rule.ex))
        ky <- c(ky, paste(rule.ex[i], rule.ex[j], sep=","))
    ky
}

as.cha <- function(cha)
{
    cha.re <- character(length=nrow(cha))
    for(i in 1:nrow(cha))
    {
      cha.re[i] <- as.character(cha[i,1])
    }
    cha.re
}

scorevec.1 <- function(indi.tr, indi.ts)
{
    score.1 <- rep(0,length(indi.ts))
    rule.ext <- names(indi.tr)[indi.tr!=0]
    nrule<-length(rule.ext)
```

```
  qq <- comb(rule.ext)

  for(i in 1:length(indi.ts))
  {
    if(sum(rslt[,2] %in% names(indi.ts)[i]) > 0)
    {
      rslt.ex <- rslt[rslt[,2] %in% names(indi.ts)[i],]
      q<-vector()
      for(j in 1:length(qq))
      {
        w <- which(match(as.cha(rslt.ex[1]), qq[j]) != "NA")
        q <- c(q,w)
      }
      rslt.ex1 <- rslt.ex[q,]
      if(length(q) > 0) score.1[i] <- max(rslt.ex1[,4])
    }
  }
  score.1
}

####시뮬레이션(20번)####
simul.hr <- vector()
simul.enet.hr <- vector()
simul.hr.1 <- vector()
simul.enet.hr.1 <- vector()

for(k in 1:20){

####train, test 분할####
  n=nrow(temp)
  tr = sample(1:n,size=floor(7*n/10),replace=F)
  temp.tr=temp[tr,]
  temp.ts=temp[-tr,]
```

```
#### apriori 알고리즘 ####
  rules=apriori(temp.tr, parameter=list(supp=0.01, conf=0.1, target="rules"))
  rules=as(rules,"data.frame")
  n=nrow(rules)
  p=NULL;
  for(i in 1:n){
    pp = gsub("\\}|\\{","",str_split(rules[i,]$rules," => ")[[1]])
    p = c(p,pp)
  }

  tmp <- t(matrix(p,nrow=2))
  rslt <- cbind(tmp,rules[,c(2,3)])
  colnames(rslt)[c(1,2)] <- c("from","to")

    for(i in 1:nrow(temp.ts))
  {
    if(i == 1) result <- t(data.frame(scorevec(temp.ts[i,],temp.ts)))
    if(i > 1) result <- rbind(result, scorevec(temp.ts[i,],temp.ts))
  }
  rownames(result)[1] <- ""

  ##hit ratio
  hr <- vector()
  for(i in 1:nrow(result))
  {
    z <- colnames(temp.ts)[temp.ts[i,] == 1]
    x <- colnames(temp.ts)[sort(result[i,], decreasing=T,index.return=T)[[2]]][1:10]
    hr[i] <- sum(x %in% z)
  }

  mean(hr)
####elastic net model####
  fit=list()
  beta=list()
```

```
  xbeta=list()
  score=matrix(0,nrow=2951,ncol=150)

  for(i in 1:150)
  {
    x.data <- temp.tr[,-i]
    y.data <- temp.tr[,i]
    fit[[i]] <- glmnet(x.data,y.data,family="binomial",lambda = 0.2,alpha=0.01)
    beta[[i]]=coef(fit[[i]])
    x.ts = temp.ts[,-i]
    x = cbind(1,x.ts)
    xbeta[[i]]=x%*%beta[[i]]
    score[,i]= as.vector(exp(xbeta[[i]])/(1+exp(xbeta[[i]])))
  }
 beta[[1]]

##hit ratio
  enet.hr <- vector()
  for(i in 1:nrow(temp.ts)){
    z <- colnames(temp.ts)[temp.ts[i,] == 1]
    x <- colnames(temp.ts)[sort(score[i,], decreasing=T,index.return=T)[[2]]][1:10]
    enet.hr[i] <-sum(x %in% z)
  }
####apriori 적용(item 분할)####
####test data item분할####
  n=ncol(temp.ts)
  ab <- which((apply(temp, 2, sum) %in% sort(apply(temp, 2, sum),
decreasing=T)[1:50]) == "TRUE")##상위 50개
  n.tr<- c(1:n)[-ab]##나머지 100개

  tr.item = sample(n.tr, size=50,replace=F)
  ab <- c(ab, tr.item)
  ab <- sort(ab)
  temp.ts.tr = temp.ts[,ab]
```

```
  temp.ts.ts = temp.ts[,-ab]

    for(i in 1:nrow(temp.ts.ts))
  {
    if(i == 1) result.1 <- t(data.frame(scorevec.1(temp.ts.tr[i,], temp.ts.ts[i,])))
    if(i > 1) result.1 <- rbind(result.1, scorevec.1(temp.ts.tr[i,], temp.ts.ts[i,]))
  }
  rownames(result.1)[1] <- ""

  ##hit ratio
  hr.1 <- vector()
  for(i in 1:nrow(result.1))
  {
    z <- colnames(temp.ts.ts)[temp.ts.ts[i,] == 1]
    x <- colnames(temp.ts.ts)[sort(result.1[i,], decreasing=T,
index.return=T)[[2]]][1:5]
    hr.1[i] <- sum(x %in% z)
  }

####elastic net model(item분할)####
  temp.tr.tr = temp.tr[,ab]
  temp.tr.ts = temp.tr[,-ab]
  temp.ts.tr = temp.ts[,ab]
  temp.ts.ts = temp.ts[,-ab]

  fit.1=list()
  beta.1=list()
  xbeta.1=list()
  score.1=matrix(0,nrow=2951,ncol=50)

  for(i in 1:50)
  {
    x.data.1 <- temp.tr.tr
    y.data.1 <- temp.tr.ts[,i]
```

```
    fit.1[[i]] <- glmnet(x.data.1,y.data.1,family="binomial",lambda = 0.2,alpha=0.01)
    beta.1[[i]]=coef(fit.1[[i]])

    x.ts.1 = temp.ts.tr
    x.1 = cbind(1,x.ts.1)
    xbeta.1[[i]]=x.1%*%beta.1[[i]]

    score.1[,i]= as.vector(exp(xbeta.1[[i]])/(1+exp(xbeta.1[[i]])))
  }

  ##hit ratio
  enet.hr.1 <- vector()
  for(i in 1:nrow(temp.ts.ts)){
    z <- colnames(temp.ts.ts)[temp.ts.ts[i,] == 1]
    x <- colnames(temp.ts.ts)[sort(score.1[i,],
decreasing=T,index.return=T)[[2]]][1:5]
    enet.hr.1[i] <-sum(x %in% z)
  }

  simul.hr[k] <- mean(hr)
  simul.enet.hr[k] <- mean(enet.hr)
  simul.hr.1[k] <- mean(hr.1)
  simul.enet.hr.1[k] <- mean(enet.hr.1)

}
```