R을 이용한 데이터마이닝

박창이[1] 김진석[2]

2008년 12월 11일

[1]서울시립대학교 통계학과 E-mail:park463@uos.ac.kr
[2]동국대학교 정보통계학과 E-mail:jinseog.kim@gmail.com

차 례

제 1 장

데이터마이닝의 기본 개념

제 1 절 데이터 마이닝이란 ?

- 정의

  대용량의 데이터로부터 이들 데이터 내에 존재하는 관계, 패턴, 규칙 등을 탐색하고 모형화함으로써 유용한 지식을 추출하는 일련의 과정들

- 배경
  - 자료의 효율적 저장을 위한 기술 (데이터 베이스, 압축, 통신)의 발달에 의한 방대한 양의 데이터 집적
  - 지식정보화 사회에서는 새로운 지식의 습득이 경쟁력의 원천 (예: 유전자 정보, 고객 정보 등)
  - 거대한 데이터의 분석을 통하여 새로운 지식의 발견가능
  - 컴퓨터 성능의 향상으로 인하여 거대한 데이터의 실시간 분석 가능

데이터 마이닝의 활용분야

- 데이터베이스 마케팅 (Database Marketing)
  - 데이터를 분석하여 획득한 정보를 이용하여 마케팅 전략 구축
  - 예: 목표마케팅 (Target Marketing), 고객 세분화 (Segmentation), 고객성향변동분석 (Churn Analysis), 장바구니 분석 (Market Basket Analysis)

- 신용평가 (Credit Scoring)
  - 특정인의 신용상태를 점수화하는 과정
  - 신용거래 대출한도를 결정하는 것이 주요 목표
  - 이를 통하여 불량채권과 대손을 추정하여 최소화함
  - 예: 신용카드, 주택할부금융, 소비자 / 상업 대출
- 생물정보학 (Bioinformatics)
  - 지놈(Genome)프로젝트로부터 얻은 방대한 양의 유전자 정보로부터 가치 있는 정보의 추출
  - 응용분야: 신약개발, 조기진단, 유전자 치료
- 텍스트 마이닝 (Text Mining)
  - 디지털화된 자료 (예: 전자우편, 신문기사 등)로 부터 유용한 정보를 획득
  - 응용분야: 자동응답시스템, 전자도서관, Web surfing
- 부정행위 적발 (Fraud Detection)
  - 고도의 사기행위를 발견할 수 있는 패턴을 자료로부터 획득
  - 응용분야 : 신용카드 거래사기 탐지, 부정수표 적발, 전화카드거래사기, 부당 / 과다 보험료 청구 탐지

데이터마이닝의 특징

- 대용량의 관측 가능한 (주로 비계획적으로 수집된) 자료를 다룸
- 컴퓨터 중심의 기법
- 경험적 방법이 중시됨
- 일반화 (generalization) 또는 예측이 중요 : 현재의 자료보다 미래의 자료를 잘 설명할 수 있는 모형을 추구
- 통계학과 컴퓨터공학(특히 인공지능)에서 함께 방법론을 개발하고 이를 경영, 경제, 정보기술(IT)분야에서 사용

데이터마이닝 관련 분야

- KDD (Knowledge Discovery in Database)
  - 데이터베이스안에서의 지식발견
  - 데이터마이닝과 가장 유사
  - KDD는 지식을 추출하는 전 과정 (계획, 자료 획득, 분석, 해석 등)을 의미하고 데이터마이닝은 KDD의 한 과정(자료의 분석)임
  - 데이터 웨어하우징 (data warehousing), OLAP (On-Line Analytical Processing) 등도 KDD의 한 과정
- 기계학습 (Machine Learning)
  - 인공지능 (Artificial intelligence)의 한 분야
  - 입력되는 자료를 바탕으로 기계(컴퓨터)가 판단을 할 수 있는 방법에 대한 연구
- 패턴인식 (Pattern Recognition)
  - 거대한 자료로부터 일정한 패턴을 찾아가는 과정
  - 이미지 분류와 깊은 관련이 있음
  - 통계학의 판별 및 분류 분석
- 통계학
  - 많은 데이터마이닝 기법들이 통계학적 관점에서 비선형 함수추정 문제임
  - 예: 신경망 모형 대 Projection Pursuit Regression

데이터 마이닝 업계 표준

Cross-Industry Standard Process for Data Mining (CRISP-DM)

- 비즈니스 이해
- 데이터 이해

- 데이터 준비
- 모델링
- 평가
- 전개

제 2 절 데이터 마이닝의 주요 모델링 기법

- 신경망 (Neural Network)
- 나무형 분류 / 회귀 (Tree-structured Model, Decision Tree): CART, C5.0, CHAID 등
- K-평균 군집화 (K-means Clustering)
  예 : 고객 세분화
- 연관성 규칙 (Association Rule): 장바구니 분석 (amazon.com)

제 3 절 데이터마이닝 적용사례

- 소매업
  - 미국의 할인점 Wall Mart 에서 매장내의 상품들과 고객들의 구매패턴의 연관성을 발견하기 위하여 연관성 분석 알고리즘을 사용
  - 기저귀와 맥주가 강한 연관성을 나타냄
  - 기저귀와 맥주를 가까이 배치하여 매출이 증가
- 신용카드회사
  - 국내의 한 신용카드회사가 부정행위를 적발하고 이를 예방하기 위한 모형의 구축
  - 기존의 카드 소지자의 구매패턴을 분석하여 현재의 구매패턴이 카드 소지자의 구매패턴과 틀린 경우 부정사용으로 의심

  - 의사결정나무와 신경망 모형이 사용됨
  - 카드의 부정사용 방지를 통하여 고객의 자산 보호 및 회사의 손해액 감소

- 의료분야
  - 종양의 악성 / 양성 판단에 의한 암 진단의 정확성을 높이기 위한 판별 및 분류분석 시행
  - 과거의 환자들에 대해서 종양검사의 결과를 근거로 (즉, 종양의 크기, 모양, 색깔 등) 종양의 악성 / 양성 여부를 구별하는 분류모형을 만든 후, 새로운 환자에서 얻은 입력변수를 이용하여 암을 진단
  - 지도학습방법 (신경망, 로지스틱 회귀모형, 의사결정나무 등) 이 사용

- 제조업
  - 반도체회사에서 불량품 자동검색장치 개발
  - 연관성 분석과 군집분석 알고리즘을 사용
  - 정상인 반도체를 그 특성에 기반하여 몇 개의 군집으로 나눈 후, 새로운 제품이 정상제품의 군집의 범위밖에 있는 경우 불량으로 규정
  - 불량품 감소로 인한 이익의 증대

- 통신회사
  - 미국의 한 장거리 회사의 23% 의 고객이 매년 이탈
  - 새로운 고객 한명을 유치하는데 필요한 비용이 $350
  - 고객성향변동관리 (churn management) 와 군집분석 (clustering) 을 이용하여 이탈의 원인을 파악 현재 고객의 40% 가 이탈 가능성이 높음
  - 이익분석 (profit analysis) 를 통하여 이탈가능성이 높은 고객을 상대로 한 마케팅이 효과적임이 입증
  - 무료 통화서비스 등의 목표마케팅 (target marketing) 으로 이탈고객 감소와 이를 통한 이익의 증가

제 4 절 데이터마이닝 솔루션들

- SAS Enterprise Miner (E-miner)
- SPSS Clementine
- IBM Intelligent Miner (I-miner)
- Oracle Darwin
- Salford CART & MARS

제 5 절 지도학습과 비지도학습

지도학습 (Supervised Learning)

- 회귀 및 분류 모형 (regression and classification)
  - 회귀 : 연속형 출력변수
  - 분류 : 범주형 출력변수
- 입력 및 출력변수의 값을 이용하여 주어진 입력변수에 대한 출력변수의 값을 예측하는 모형을 개발
- 기법: 판별분석, 회귀분석, 로지스틱 회귀분석, 의사결정나무, 신경망 등
- 예 : 특정 기업의 정보(재무제표 등)을 이용하여 1년 후의 회사의 파산 여부를 예측

비지도학습 (Unsupervised Learning)

- 출력변수가 없음
- 입력변수간의(혹은 내의) 관계를 탐색적으로 분석하여 의미 있는 정보 추출
- 군집 분석, 연관성 분석, 주성분 / 인자분석

- 예 : 한국 성인 남자의 골격을 몇 개의 그룹으로 나눈 후 기성복 사이즈의 종류를 결정

지도학습모형

$X_1, \ldots, X_p$: 입력변수, Y: 출력변수

- 회귀 모형: 연속형 출력변수

$$Y = f(X_1, \ldots, X_p) + \varepsilon$$

  평가기준: $MSPE = E[Y - f(X_1, \ldots, X_p)]^2$

- 분류 모형: 범주형 출력변수

$$\mathbb{P}(Y = j | X_1, \ldots, X_p) = f(X_1, \ldots, X_p)$$

  평가기준: $\mathbb{P}(Y \neq j | X_1, \ldots, X_p), j = 1, \ldots, k$

제 6 절 모형의 평가

- 하나의 자료 분석시 여러가지 가능한 모형을 적합시키게 되는데 이중 최적의 모형을 선택하기 위해 필요
- 모형의 평가 방법
  - 예측력: 얼마나 잘 예측하는가?
  - 해석력: 모형이 입력/출력 변수간의 관계를 잘 설명하는가?
  - 효율성: 얼마나 적은 수의 입력변수로 모형을 구축했는가?
  - 안정성: 모집단의 다른 자료에 적용했을 때 같은 결과를 주는가?
- 모형의 평가: 어떤 모형이 랜덤하게 예측하는 모형보다 예측력이 우수한지, 그리고 고려된 모형들 중 어느 모형이 가장 좋은 예측력을 보유하고 있는지를 비교 / 분석

학습오차와 예측오차

- 오차
  - 학습오차: 학습자료로부터 구한 오차
  - 예측오차: 미래의 자료로부터 구한 오차
- 지도학습은 일반화에 관심을 둔다. 따라서, 학습오차보다는 예측오차에 더 많은 관심을 둔다. 즉, 지도학습의 목적은 예 측오차를 최소화하는 모형의 구축에 있다

모형의 복잡도

- 예
  - 입력변수를 많이 사용할 수록 모형이 복잡해짐
    모형 $y = a + b_1x_1 + b_2x_2$ 가 모형 $y = a + b_1x_1$ 보다 복잡
  - 입력변수와 출력변수의 관계를 나타내는 식이 비 선형적일수록 모형이 복잡
    모형 $y = \dfrac{1}{1 + \exp(a + bx)}$ 가 모형 $y = a + bx$ 보다 복잡

과적합 (Overfitting)

- 정의: 매우 복잡한 모형을 사용하여 학습오차를 매우 작게 한 경우 예측오차가 매우 커질 수 있는데, 이러한 현상을 과적합이라 한다.
- 학습오차는 사용되어진 자료를 통하여 구할 수 있으나, 예측오차는 미래의 자료에 대한 오차로서 실제로는 구할 수 없다.
- 따라서 지도학습에서 학습오차를 너무 작게 하는 것이 항상 좋은 것은 아니다.
- (예) 개인 신용평가
  - 목적: 은행 고객의 수입과 나이의 정보로 그 고객의 미래 신용 상태를 예측
  - 입력: 수입, 나이
  - 출력 : 신용 상태 (양호 / 불량)

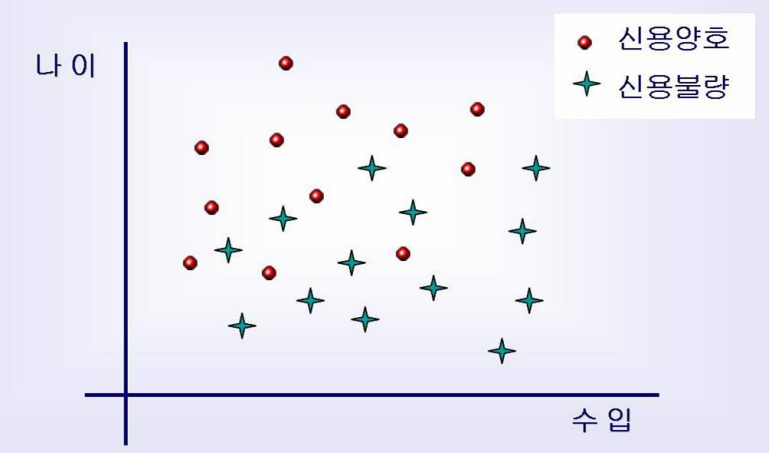


그림 1.1: 개인 신용평가 예제: 훈련자료

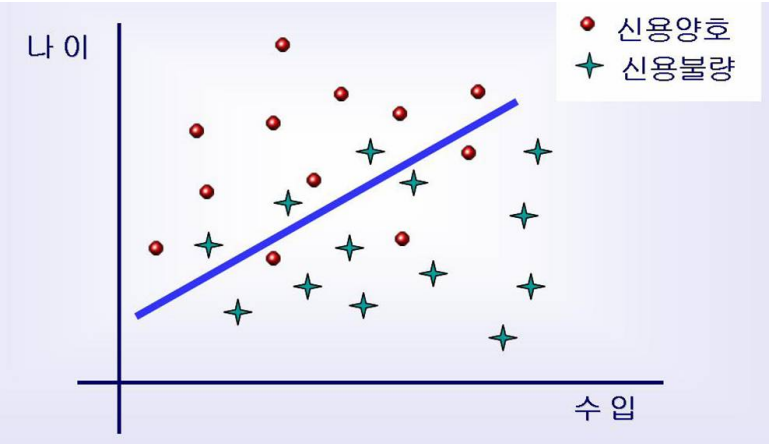


그림 1.2: 개인 신용평가 예제: 단순한 모형

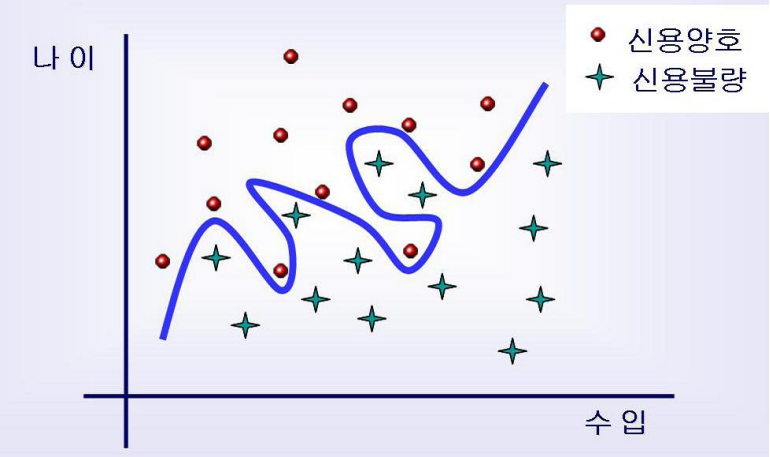


그림 1.3: 개인 신용평가 예제: 복잡한 모형

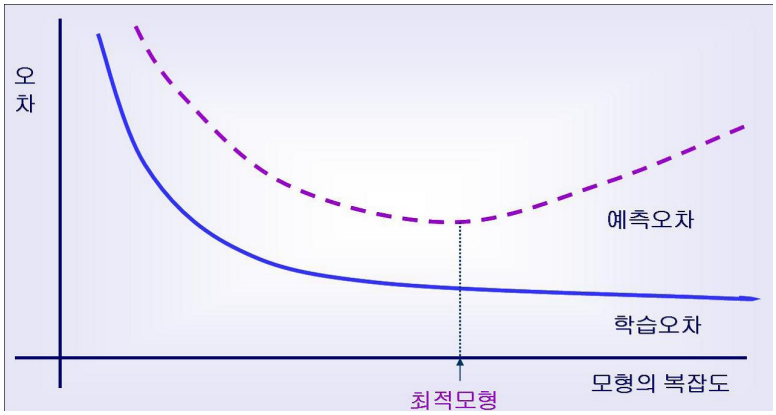


그림 1.4: 학습오차, 예측오차, 모형의 복잡도

모형의 예측력 평가

- 데이터 분할 (data partitioning) 방법
  - 전체 자료를 훈련자료 (training data) 와 테스트자료 (test data) 로 나누고 훈련자료를 이용하여 모형 적합하고 적합된 모형을 테스트 자료에 적용하여 예측하며 테스트 자료에 대하여 평가 기준을 이용하여 평가
  - 보통 특정한 분할에 의한 bias 를 줄이기 위해 데이터에 대한 랜덤한 분할을 여러 번 반복 (repetition) 실시한 후 평가기준 값의 평균을 냄
  - 방법에 따라 훈련, 검증 (validation), 테스트 자료로 나누어서 검증 데이터를 이용하여 모형을 최적화 하는 경우도 있음
- Penalization (Regularization) 방법
  - 학습자료를 이용하여 다양한 복잡도를 갖는 모형을 구축
  - 각 모형의 학습오차에 모형의 복잡도에 대한 적절한 penalty 를 더해 비용함수 생성 (예) AIC, BIC 에서 입력변수의 갯수
  - 비용함수를 최소화하는 모형 선택
- 교차 확인 (Cross Validation, CV) 방법
  - 데이터 분할 방법의 일반화로 볼 수 있음
  - k-fold CV 알고리즘

1. 데이터를 k 개의 셋으로 분할
2. 각 j 에 대해 j 번째 셋을 제외한 나머지를 이용하여 모형 적합
3. j 번째 셋을 테스트 자료로 이용하여 예측오차를 구함
4. k 개의 예측오차의 평균을 예측오차로 활용하여 최적 모형 선택 여기서 $j = 1, \ldots, k$.

분류모형의 평가

오분류표

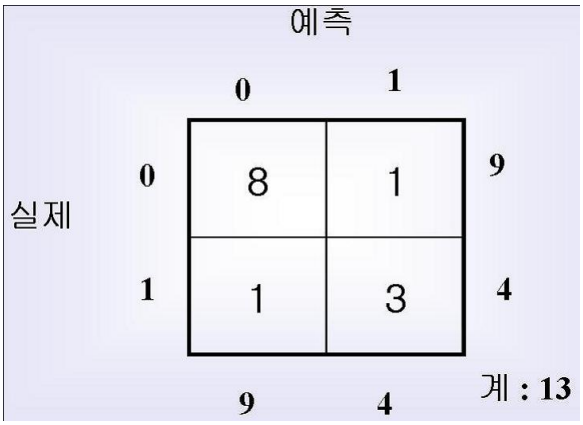


그림 1.5: 오분류표 예제

- 오분류률에 대한 다양한 측도
  - 정분류율=(실제 0, 예측 0) 빈도+(실제 1, 예측 1) 의 빈도 / 전체빈도
    (예) 정분류율 = (8+3)/13 = 11/13
  - 오분류율=(실제 0, 예측 1) 빈도+(실제 1, 예측 0) 의 빈도 / 전체빈도
    (예) 오분류율 = 1 - 11/13 = 2/13
  - 민감도 (sensitivity)=(실제 1, 예측 1) 의 빈도 / 실제 1 의 빈도
    (예) 민감도 = 3/4
  - 특이도 (specificity)=(실제 0, 예측 0) 의 빈도 / 실제 0 의 빈도
    (예) 특이도 = 8/9
- 민감도는 1 에서의 정분류율로 특이도는 0 에서의 정분류율

ROC 도표

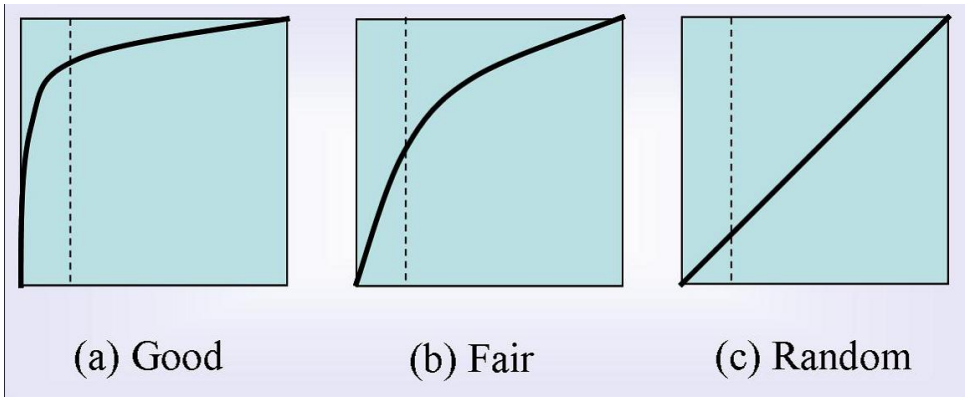


그림 1.6: ROC 도표 예제

- ROC : Receiver Operator Characteristic
- 여러 개의 분류 기준값에서 구하여진 민감도 특이도 값의 쌍들을 x 축에는 1-특이도, y 축에는 민감도를 지정하고 그린 곡선
- ROC 곡선의 형태
- 기타 : Lift chart, response threshold chart 등

제 2 장

선형회귀모형

제 1 절 단순선형회귀

예제 : 에어컨 판매대수 예측

- 목적 : 에어컨 예약대수를 이용하여 내년에 판매되는 에어컨 대수를 예측
- 자료 : 지난 10 년간의 에어컨 예약대수와 판매대수
- 입력변수 : 에어컨 예약대수
- 목적변수 : 에어컨 판매대수

단순선형회귀 모형

- 모형 : $y = a + bx + \varepsilon$
  x: 에어컨 예약대수, y: 에어컨 판매대수, ε: 오차항 (평균이 0 이고 분산이 σ^2)
- 선형회귀모형 : 입력변수와 출력변수의 관계가 선형방정식
  참고 : 비선형회귀모형 (예 : $y = \sin(2x) + \varepsilon$)
- 단순 선형회귀모형 : 입력변수가 하나인 선형회귀모형
- 다중 선형회귀모형 : 입력변수가 2 개 이상인 선형회귀모형

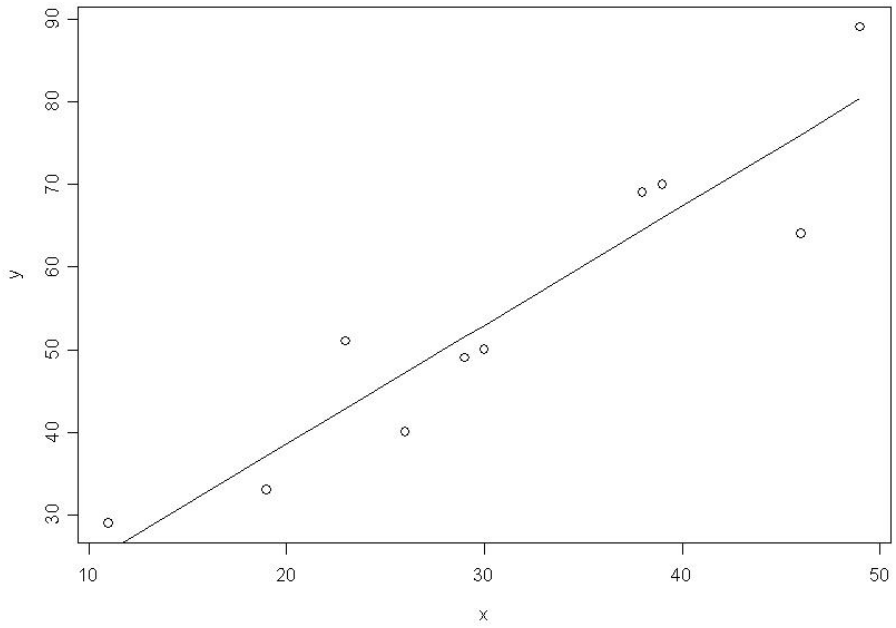


그림 2.1: 에어컨 판매대수 자료

모수의 추정

- 모수: (a, b)
- 최소제곱추정치: $(\hat{a}, \hat{b})$

  오차제곱합 $\sum_{i=1}^{n}(y_i - a - bx_i)^2$ 을 최소로 하는 (a, b) 로

$$\begin{aligned} \hat{b} &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\ \hat{a} &= \bar{y} - \hat{a}\bar{x} \end{aligned}$$

예측 및 모형의 해석

- 예측

  주어진 새로운 입력변수 x 에 대하여 출력변수 y 를 $\hat{y}$ 로 예측. 여기서 $\hat{y} = \hat{a} + \hat{b}x$
- 모형의 해석
  - x 가 한 단위 증가할 때의 y 의 증가량

- b 가 양수이면 x 가 증가하면서 y 가 증가
- b 가 음수이면 x 가 증가할 때 y 는 감소

예제의 결과

- 최소제곱법으로 추정된 모형식

$$y = 9.74 + 1.44x$$

- 예측
  올해 에어컨 예약대수가 45 이면 내년 에어컨 판매대수는 $9.74 + 1.44 * 45 = 74.54$ 로 예측
- 해석
  에어컨 예약대수가 1 단위 증가하면 에어컨 판매대수는 1.44 단위 증가

회귀계수의 검정

- 회귀계수 b 가 0 이면 입력변수 x 와 출력변수 y 사이에 아무런 관계가 없다. 즉, 회귀계수 b 가 0 이면 적합 된 추정식은 아무 의미가 없게 된다.
- 적합 된 추정식이 의미가 있는지(자료를 잘 설명하는지)를 검정하는 것은 회귀계수 b 가 0 인지를 검정하는 것과 같음
- 검정통계량

$$t = \frac{\hat{b}}{s.e.(\hat{b})}$$

- 검정방법
  $|t|$ 가 크면 b 가 0 이라는 가설(귀무가설)을 기각한다. 즉, 추정된 회귀식이 유의하다라고 결론을 내림

회귀계수 검정 예제

Variable	df	Estimate	SE	T	prob>|t|
Intercep	1	9.7364	6.6620	1.471	0.1796
X	1	1.4407	0.2004	7.188	0.0001

- x 에 대한 유의확률 (p-value) 이 매우 작으므로 (0.001) 회귀계수 b 가 0 이라는 가설은 기각
- 추정된 회귀직선은 자료를 잘 설명함

제곱합의 분할과 결정계수

- 제곱합의 분할: 전체제곱합 = 회귀제곱합 + 잔차제곱합
  - 전체제곱합: $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
  - 회귀제곱합: $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$
  - 잔차제곱합: $\sum_{i=1}^{n}(y_i - \bar{y})^2$
- 결정계수: $R^2 =$ 회귀제곱합 / 전체제곱합
  - $0 \leq R^2 \leq 1$
  - 1 에 가까울 수록 회귀모형이 자료를 잘 설명
  - 에어컨 판매대수 예제
    $R^2 = 0.8659$: 자료의 전체 변동중 회귀모형이 86.59% 를 설명

선형회귀모형에 사용되는 가정들

- 선형회귀모형에 사용되는 가정
  - 선형성: 입력변수와 출력변수와의 관계가 선형적
  - 등분산성: 오차의 분산이 입력변수와 무관하게 일정
  - 정규성: 오차의 분포가 정규분포
- 선형모형을 자료에 적합하기 전에 위의 3 가지 가정이 만족되는지를 확인

- 선형성은 위의 가정 중 가장 중요한 가정으로 이 가정이 맞지 않은 경우에는 선형회귀모형은 좋지 않은 결과를 제공

- 가정들의 검증 방법

  - 선형성: 단순선형회귀모형에서는 입력변수와 출력변수와 산점도를 이용하고 다중선형회귀모형에서는 잔차와 출력변수와의 산점도를 이용
  - 선형회귀모형에 사용된 모든 가정이 만족되는 경우에는 이 잔차는 아무런 정보가 없는 오차와 비슷할 것임
  - 따라서, 잔차의 산점도에서 어떤 패턴이 발견되면, 선형회귀모형의 가정을 의심

- 가정들이 맞지 않은 경우

  - 선형성의 가정이 만족되지 않은 경우
    다항회귀 (polynomial regression) 등의 비선형회귀모형이나, 회귀모형에 아무런 가정도 하지 않는 비모수 회귀모형 (예: 의사결정나무, 신경망 모형) 등을 사용
  - 등분선성이 만족하지 않는 경우
    가중회귀방법 (weighted regression) 을 사용
  - 정규성이 만족하지 않는 경우
    로버스트 회귀방법 (robust regression) 을 사용

제 2 절 다중선형회귀

다중선형회귀 모형

- 입력변수가 2 개 이상인 선형회귀모형
- 입력변수: $\boldsymbol{x} = (x_1, \ldots, x_p) \in \mathbb{R}^p$
- 출력변수: $y \in \mathbb{R}$

- 모형

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

  ε 은 오차항으로 평균이 0 이고 분산이 σ^2

- 모수 (회귀계수): $(\beta_0, \beta_1, \ldots, \beta_p)$

다중선형회귀 : 회귀계수의 추정 및 검정

자료 : $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, 여기서 $\boldsymbol{x}_i = (x_{1i}, \ldots, x_{pi})$

- 추정
  - 최소제곱추정치 : $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$
  - 오차제곱합 $\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_{1i} - \cdots - \beta_p x_{pi})^2$ 을 최소로 하는 $(\beta_0, \beta_1, \ldots, \beta_p)$
- 유의성 검정
  - 검정통계량 : $t_i = \frac{\hat{\beta}_i}{s.e.(\hat{\beta}_i)}$
  - $|t_i|$ 가 크면 회귀계수 β_i 는 유의하다, 즉, β_i 가 0 이 아니다라고 결론

예측 및 모형의 해석

- 예측

  주어진 새로운 입력변수 $\boldsymbol{x} = (x_1, \ldots, x_p)$ 에 대하여 출력변수 y 를 $\hat{y}$ 로 예측. 여기서 $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots \hat{\beta}_p x_p$

- 모형의 해석
  - x_i 가 한 단위 증가할 때의 y 의 증가량
  - $\hat{\beta}_i$ 가 양수이면 x_i 가 증가하면서 y 가 증가
  - $\hat{\beta}_i$ 가 음수이면 x_i 가 증가할 때 y 는 감소

ANOVA 표 및 모형의 유의성 검정

요인	제곱합	자유도	제곱평균	F-값
회귀	회귀제곱합 (SSR)	p	MSR = SSR/p	F =MSR/MSE
잔차	잔차제곱합 (SSE)	n-p-1	MSE =SSE/(n-p-1)	
계	전체제곱합 (SST)	n-1		

F-값이 크면 모형이 유의하다고 결론

다중선형회귀 예제

A 건설회사가 고객에게 주로 받는 질문은 겨울에 난방비가 얼마나 나오는가 하는 것이다. 이 질문에 답하기 위하여 A 건설회사는 20 개의 아파트를 임의로 추출하여 다음의 3 가지 항목을 조사하였다.

- 겨울의 평균온도
- 방열재의 두께
- 아파트 나이
- 회귀계수 추정값

입력변수	회귀계수	표준오차	T	유의확률
절편	427.19	59.60	7.17	0.000
온도	-4.58	0.77	-5.93	0.000
방열재	-14.83	4.75	-3.12	0.007
나이	6.10	4.01	1.52	0.14

- ANOVA 표

요인	제곱합	자유도	제곱평균	F-값	유의확률
회귀	171220	3	57073	21.9	0.000
잔차	41695	16	2606		
계	212916	19			

- 회귀추정식

$$\hat{y} = 427.19 - 4.58x_1 - 14.83x_2 + 6.10x_3$$

- 해석
  - 평균온도 1 도가 떨어지면 난방비가 4.58 단위 증가
  - 방열재의 두께가 1 단위 증가하면 난방비가 14.83 단위 감소
  - 아파트의 나이가 1 년 증가하면 난방비가 6.10 단위 증가

제 3 절 다항회귀모형

다항회귀 예제

예제: 다음의 자료는 특정한 화학제품을 만드는데 있어서 촉매와 산출량과의 관계를 규명하기 위한 것이다.

촉매	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
산출량	10.2	13.71	19.6	23.5	25.6	23.3	25.1	17.5	14.3	0.02

다항회귀 모형

- 다항회귀모형 (polynomial regression model)
  출력변수와 입력변수와 관계가 선형이 아니지만 이차함수로 잘 적합할 수 있음

$$y = \beta_0 + \beta_1 x^1 + \cdots + \beta_p x^p + \varepsilon$$

- 모수의 추정
  $z_1 = x, z_2 = x^2, \ldots, x_p = x^p$ 라 놓고 이 새로운 p 차원의 입력변수와 출력변수 사이의 다중선형회귀모형 적합

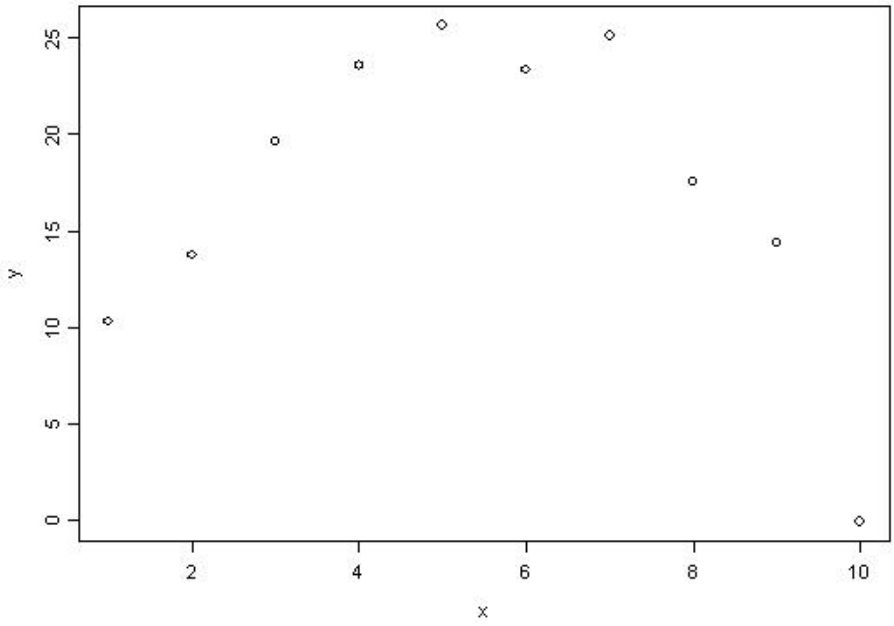


그림 2.2: 다항회귀 예제 자료

차수 p의 결정

- $p = 2$부터 하나씩 증가시키면서 모형을 적합
- p가 증가하면서 결정계수값은 계속해서 증가
- 결정계수값의 증가가 둔화되는 p를 선택
- 다항회귀모형은 차수의 결정에 따라 그 결과가 민감하게 반응하는 경향이 있다. 따라서, 차수(p)의 결정은 매우 신중하게 결정해야 함

다항회귀 예제 분석

- 차수의 결정: 2차 함수가 적합
- 회귀계수의 추정

입력변수	회귀계수	표준오차	T	유의확률
절편	-1.26	2.71	-0.46	0.65
x	10.32	9.11	9.11	0.00
x^2	-0.99	0.10	9.88	0.00

- 이차식의 회귀계수가 유의하므로 이차함수가 적합

범주형 입력변수

- 범주의 수가 J 개인 경우 $(J-1)$ 개의 가변수가 필요
- 입력변수가 k 번째 범주에 속하는 경우에 $x_k = 1$ 로, 그리고 나머지 $(J-2)$ 개의 x 는 0 으로
- 예: $J = 3$

범주	x_1	x_2
1	1	0
2	0	1
3	0	0

예제: 다음 자료는 임의로 선택된 10 명에 대한 집소유 여부와 월 소득과의 관계를 나타낸 것이다.

집소유	소	소	미	미	미	미	미	미	소	소
월소득	39	49	22	41	28	21	30	47	23	50

소: 소유, 미: 미소유

- 가변수를 이용
- 범주가 두개인 경우에는 입력변수가 첫번째 범주인 경우에 x 를 0 으로 입력변수가 두번째 범주인 경우에는 x 를 1 로 놓음

집소유 (x)	0	0	1	1	1	1	1	1	0	0
월소득 (y)	39	49	22	41	28	21	30	47	23	50

소유 :0, 미소유 : 1

- 모형 : $y = a + bx$
- 회귀계수의 추정

입력변수	회귀계수	표준오차	T	유의확률
절편	40.25	5.63	7.14	0.00
x	-8.75	7.27	-1.20	0.26

- 해석 : 집을 소유한 사람이 미소유인 사람보다 소득이 8.75 단위 높다. 그러나 이 차이는 통계적으로 유의하지 않음

제 4 절 변수 선택

필요한 이유

- 많은 입력변수를 관리하는데 필요한 노력과 비용의 절약
- 다중공선성에 의한 예측치의 분산의 증가 방지
- 적절한 모형의 복잡도 (변수의 개수) 유지로 예측오차의 감소가능
- 유의한 입력변수를 모형에 넣지 않으면, 중요한 정보를 놓칠 수 있으며, 또한 그 결과가 bias 된다. 따라서, 변수선택에 있어서는, 너무 많은 변수를 사용해서 생기는 문제를 피하고, 동시에 중요한 변수를 놓쳐서 생기는 정보의 손실을 최소로 해야 한다. 즉, 적당한 수의 변수를 선택해야 한다.

Best Subset Selection

기본과정

- p 개의 입력변수가 있는 경우 $(p+1)$ 개의 모형 $M_0, M_1, \ldots, M_p$ 를 고려. 여기서 M_k 는 k 개의 변수만을 포함하는 모형
- 주어진 $(p+1)$ 개의 모형 중 가장 좋은 모형을 선택

- 가장 좋은 모형: 모형선택의 판정기준을 사용
- 모형선택의 기준으로는 AIC(Akaike Information Creteria), 수정된 결정계수(Adjusted coefficient of determination), Mallow's Cp 등이 사용

모형을 만드는 방법

$(p+1)$ 개의 모형 $M_0, M_1, \ldots, M_p$ 를 만드는 방법에 따라 다음과 같은 변수선택방법들이 개발되었음

- 모든 가능한 회귀 (All possible method)
  - M_k 는 k 개의 변수를 포함하는 모든 모형들 중 오차제곱합을 최소로 하는 모형
  - $M_0, M_1, \ldots, M_p$ 를 만들기 위하여는 2^p 개의 모형을 적합시켜야 함
  - p 가 큰 경우 현실적으로 사용하기 어려움
- 전진선택법 (Forward selection)
  - M_{k-1} 모형을 구축한 후 M_{k-1} 모형에 사용되지 않은 $p-(k-1)$ 개의 변수를 하나씩 모형에 넣어 오차제곱합 구함
  - 오차제곱합을 가장 작게 하는 변수를 M_{k-1} 에 넣어서 k 번째 모형 M_k 를 만듬
  - 입력변수가 p 개인 경우에 $p(p+1)/2$ 개의 모형을 적합하면 됨
  - 단점: 한번 선택된 변수는 절대로 제거되지 않음
- 후진소거법 (Backward elimination)
  - M_k 을 구축한 후 M_k 모형에 사용된 k 개의 변수를 하나씩 모형에서 제거하고 오차제곱합을 구함
  - 오차제곱합을 가장 작게 하는 변수를 M_k 에서 제거하여 $k-1$ 번째 모형 M_{k-1} 를 만듬
  - 입력변수가 p 개인 경우에 $p(p+1)/2$ 개의 모형을 적합하면 됨

  - 단점: 한번 제거된 변수는 절대로 다시 선택되지 않음

- 단계적 방법 (Stepwise method)
  - 전진선택법과 후진소거법을 결합
  - 중요한 변수를 하나씩 추가로 선택하면서 이미 선택된 변수들이 제거될 수 있는지를 매 단계마다 검토
  - 이 방법은 $(p+1)$ 개의 모형을 만드는 것이 아니라, 최적의 모형 하나만을 결과로 줌
  - 일반적으로 전진선택법이나 후진소거법보다 좋음
  - p 가 큰 경우 계산량이 매우 많을 수 있음

변수선택 기준

- 복잡한 모형은 작은 오차제곱합을 갖지만, 모형의 복잡도에 따른 예측오류 증가
- 모형선택 기준의 기본 아이디어는 오차제곱합과 모형복잡도(변수의 개수) 를 동시에 고려
- 기준에 따라 최종 모형이 달라질 수 있음
- 수정결정계수
  - R^2 는 변수가 많아질수록 증가
  - 수정결정계수
$$R^2_{adj} = 1 - \frac{n-1}{n-p}(1-R^2)$$
  - 변수가 많아지면 $1-R^2$ 는 줄지만 $\frac{n-1}{n-p}$ 가 증가하여 수정결정계수는 항상 증가하지는 않음
  - 최대로 하는 모형을 선택
- 멜로우의 Cp
  - Mallow's Cp = 잔차제곱합 / 분산 추정량 $+2p-n$

- 변수의 수가 증가하면 오차제곱합은 감소하지만 $2p - n$ 이 증가하고, 따라서 Mallow's Cp 는 처음에는 감소하다 모형이 복잡해지면 다시 증가
- Cp 를 최소로 하는 모형을 선택한다.

Regularization Methods*

능형회귀 (Ridge Regression)

- 능형회귀 추정량

$$\boldsymbol{\beta}^{\text{ridge}} = \text{argmin} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{k=1}^{p} x_{ik}\beta_k \right)^2$$

subject to $\sum_{k=1}^{p} \beta_k^2 \leq s$ 또는

$$\boldsymbol{\beta}^{\text{ridge}} = \text{argmin} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{k=1}^{p} x_k\beta_k \right)^2 + \lambda \sum_{k=1}^{p} \beta_k^2. \tag{2.1}$$

- 위에서 $s = 0$ 이면 모형은 상수항 (intercept term) 만을 포함한다. 반대로 $s = \infty$ 이면 최소제곱법으로 추정된 모형과 동일하다.
- 이 ridge estimator 는 Hoerl and Kennard (1970) 이 소개하였고, $p > n$ 일 때 최소제곱추정량을 계산하기 위해 고안 되었다.
- 선형회귀에서 통상적인 최소제곱추정량은 아래와 같다.

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

where $\mathbf{X} = (\mathbf{x}_1', \ldots, \mathbf{x}_n')'$ and $\mathbf{y} = (y_1, \ldots, y_n)'$.

- 하지만 $p > n$ 경우, $(\mathbf{X}'\mathbf{X})^{-1}$ 을 계산할 수가 없다(존재하지 않는다). 보통 이 경우, 다음 식을 푸는데, 이 경우에는 해가 여러개 존재한다.

$$(\mathbf{X}'\mathbf{X})\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y}$$

- 위의 식 (1) 에서 ridge estimator 는 최소제곱추정량의 $(\mathbf{X}'\mathbf{X})^{-1}$ 부분을 $(\mathbf{X}'\mathbf{X}+\lambda\mathbf{I})^{-1}$ 로 치환한 값이다. 즉,

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

또는

$$\hat{\beta}_j^{ridge} = \hat{\beta}_j/(1+\lambda), j = 1, \ldots, p.$$

LASSO: Least Absolute Shrinkage and Selection Operator

- Ridge regression 의 단점으로는 모든 input 변수들이 모형에 포함되어 있어 모형에 대한 해석이 용이하지 않다는 데 있다.
- 그렇다면 변수선택과 예측력 향상을 동시에 가능하게 하는 방법은 없을까 ?
- 놀랍게도 그런 방법이 있으며, Tibshirani (1996) 가 제안한 LASSO (Least Absolute Shrinkage and Selection Operator) 라는 방법이다.
- LASSO 방법은 다음의 목적함수를 최소화 하는 $\boldsymbol{\beta}$ 를 찾는다.

$$\sum_{i=1}^{n} l(y_i, \mathbf{x}_i'\boldsymbol{\beta}) + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{penalty function}} .$$

- ridge 와 다른 점은 penalty function 이 제곱에서 절대값으로 바뀌었다는 것밖에 없다.
- 이러한 penalty 를 l_1 penalty 라고 부른다, 반면에 Ridge 의 penalty 를 l_2 penalty 라고 함

제 5 절 R 예제 : swiss 데이터

Swiss Fertility and Socioeconomic Indicators (1888) Data

- Fertility: common standardized fertility measure
- Agriculture: % of males involved in agriculture as occupation

- Examination: % draftees receiving highest mark on army examination
- Education: % education beyond primary school for draftees.
- Catholic: % 'catholic'
- Infant.Mortality: live births who live less than 1 year

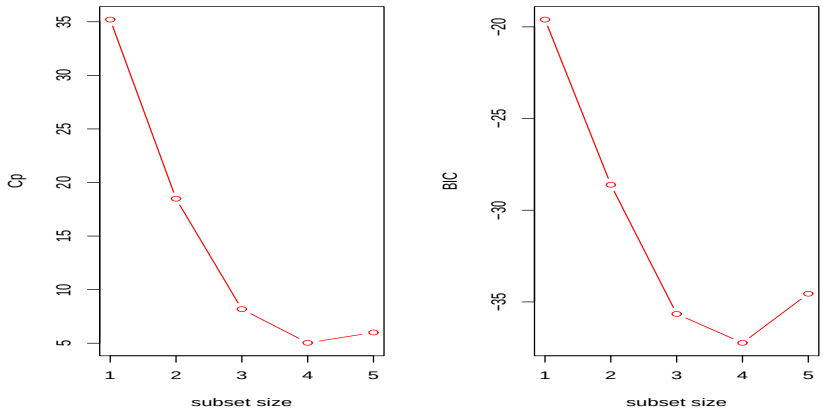


그림 2.3: Swiss 데이터 변수 선택

```
library(leaps)
data(swiss)
a = regsubsets(Fertility~., data=swiss)
summary(a)

Subset selection object
5 Variables  (and intercept)
                 Forced in Forced out
Agriculture          FALSE      FALSE
Examination          FALSE      FALSE
Education            FALSE      FALSE
Catholic             FALSE      FALSE
Infant.Mortality     FALSE      FALSE
1 subsets of each size up to 5
```

```
Selection Algorithm: exhaustive
         Agriculture Examination Education Catholic Infant.Mortality
1  ( 1 ) " "         " "         "*"       " "      " "
2  ( 1 ) " "         " "         "*"       "*"      " "
3  ( 1 ) " "         " "         "*"       "*"      "*"
4  ( 1 ) "*"         " "         "*"       "*"      "*"
5  ( 1 ) "*"         "*"         "*"       "*"      "*"
> par(mfrow = c(1,2))
> plot(1:5, summary(a)$cp, type="b", xlab="subset size", ylab="Cp", col="red" )
> plot(1:5, summary(a)$bic, type="b", xlab="subset size", ylab="BIC", col="red" )

> lm(Fertility ~ Agriculture+Education+Catholic+Infant.Mortality, data=swiss)
Call:
lm(formula = Fertility ~ Agriculture + Education + Catholic +
    Infant.Mortality, data = swiss)
Coefficients:
     (Intercept)       Agriculture         Education          Catholic
         62.1013           -0.1546           -0.9803            0.1247
Infant.Mortality
          1.0784
```

제 3 장

로지스틱 회귀분석

제 1 절 로지스틱 회귀

범주형 출력변수에 대한 회귀분석

- y 가 범주형 일때 $(y = 0, 1)$ 선형회귀모형 $y = a + bx + e$ 적용시 문제점
  - $a + bx$ 는 [0,1] 을 벗어날 수 있음
  - 오차항 e 의 분포가 정규분포가 아님
- 대안: $\mathbb{P}(Y = 1|x) = F(a + bx)$, 여기서 $F(x)$ 는 연속이고 증가하며 [0,1] 사이에서 값을 갖는 함수
- 여러 가지 $F(x)$
  - Logistic 모형: $F(x) = \exp(x)/(1 + \exp(x))$
  - Gumbel 모형: $F(x) = \exp(-\exp(x))$
  - Probit 모형: $F(x)$ 가 표준정규분포의 분포함수 (생물학 독성실험 등)
- 로지스틱 모형이 계산의 편이성으로 인하여 가장 널리 쓰임

로지스틱 모형과 오즈비

- 모형: $\mathbb{P}(Y = 1|x) = \exp(a + bx)/(1 + \exp(a + bx))$

- 오즈비 (odds ratio)의 정의

$$\frac{\mathbb{P}(Y=1|x+1)\mathbb{P}(Y=0|x)}{\mathbb{P}(Y=0|x+1)\mathbb{P}(Y=1|x)} = \exp(b)$$

- x가 한 단위 증가 할 때 $y=1$일 확률과 $y=0$일 확률의 비의 증가율
- 예
  - x는 소득, y는 어떤 상품에 대한 구입여부(1=구입,0=미구입)
  - $b = 3.72$
  - 소득이 한 단위 증가하면 물품을 구매하지 않을 확률에 대한 구매할 확률의 비(오즈비)가 $\exp(3.72) = 42$배 증가함

모수의 추정

- 최대우도(maximum likelihood)추정량
- 우도함수(likelihood function)

$$L(a,b) = \prod_{i=1}^{n} F(a+bx_i)^{y_i}(1-F(a+bx_i))^{1-y_i},$$

  여기서 $F(x) = \exp(x)/(1+\exp(x))$

- 우도함수를 최대화하는 최대우도추정량 $(\hat{a},\hat{b})$는 수치적 방법을 사용하여 구함

입력변수의 유의성 검정

- 입력변수 x가 y를 설명하는데 유의한지, 즉, 회귀계수 b가 0인지,를 검정
- 우도비 검정 통계량

$$\chi^2 = -2(\max_a l(a,0) - l(\hat{a},\hat{b})),$$

  여기서, $l(a,b) = \log L(a,b)$

- χ^2가 크면 b가 0이 아니라고 결론

- 근사적으로 χ^2 는 카이제곱분포를 따름

로지스틱 회귀의 확장

- 다중 로지스틱 회귀
  - 입력변수가 2 개 이상이고 출력변수가 범주형 자료 (범주가 2 개)
  - 모형

$$\mathbb{P}(Y = 1|\boldsymbol{x}) = F(a + b_1x_1 + \cdots + b_px_p),$$

    여기서 $F(x) = \exp(x)/(1 + \exp(x))$
  - 모수의 추정: 최대우도추정치
  - 회귀계수의 유의성 검정: 우도비 검정을 이용
- 다항 로지스틱 모형
  - 모형

$$\mathbb{P}(Y = 1|x) = F(a + b_1x + \cdots + b_px^p),$$

    여기서 $F(x) = \exp(x)/(1 + \exp(x))$
  - 모수의 추정: 최대우도추정치

범주형 입력변수

- 선형회귀모형과 마찬가지로 가변수를 사용
- 가변수를 사용한 모형에서 추정된 회귀모형에 대한 해석시 주의

변수선택

- 선형회귀모형과 마찬가지로 모든 가능한 회귀, 전진선택법, 후진소거법 그리고 단계적방법 등을 사용
- 선형회귀모형에서 사용하는 오차제곱합 대신에 로그우도함수 값을 사용

- 예를 들면, 전진선택법에서는 각 단계마다 로그우도함수값의 증가량이 가장 큰 변수를 선택
- 모형선택기준으로는 AIC(Akaike Information creterion), BIC (Bayesian Information criterion) 등 사용
- AIC 또는 BIC가 최소인 모형 선택

제 2 절 예제

- 회사채의 신용등급(안정 / 위험)과 여러 가지 재무자료들 사이의 관계
- 입력변수
  - x_1: 자산대비 부채현황 지표
  - x_2: 현금회전율
  - x_3: 종업원 수(50인 이하=0, 50-100인=1, 100인 이상=2)
- 출력변수: 회사채 신용등급 (안정=1, 위험=0)
- x_3가 질적변수이므로 가변수가 필요
- x_3를 위한 가변수 z_1과 z_2

범주	z_1	z_2
50인 이하	0	0
50-100인	1	0
100인 이상	0	1

- 가변수를 포함하는 로지스틱 회귀모형

$$\mathbb{P}(Y=1|\boldsymbol{x}) = F(a + b_1x_1 + b_2x_2 + b_3z_1 + b_4z_4),$$

  여기서 $F(x) = \exp(x)/(1+\exp(x))$

- 회귀계수 추정값

입력변수	회귀계수	표준오차	χ^2	유의확률
절편	12.221	3.594	11.562	0.000
x_1	-1.072	0.393	7.438	0.006
x_2	11.524	4.131	7.781	0.005
z_1	0.698	0.357	3.940	0.047
z_2	2.614	0.791	10.907	0.001

- 모든 변수가 작은 유의확률을 갖으므로 출력변수를 설명하는데 유의
- x_1 의 회귀계수의 부호가 음수이므로 부채비율이 높을 수록 회사채의 신용이 낮음을 의미하고 x_2 의 회귀계수의 부호가 양수이므로 현금회전율이 높을 수록 회사채의 신용이 높아짐
- z_1: 50-100 인 규모의 회사와 50 인 미만의 회사의 회사채 신용등급의 오즈비는 $\exp(0.6981) = 2.00$ 으로, 50-100 인 규모의 회사의 화사채가 50 인 미만의 회사채에 비하여 약 2 배정도 신용이 좋다는 의미
- z_2: 100 인 이상 규모의 화사와 50 인 미만의 회사의 회사채 신용등급의 오즈비는 $\exp(2.6141) = 13.65$ 으로 100 인 이상 규모의 회사의 화사채가 50 인 미만의 회사채에 비하여 약 13.65 배정도 신용이 좋다는 의미
- 가변수인 z_1 과 z_2 의 해석은 항상 50 인 미만의 회사(가변수 값이 모두 0 인)와 비교
- 가변수를 다르게 만들면 회귀계수의 추정치가 달라지나 오즈비는 항상 일정함. 따라서, 그 결과의 해석에는 차이가 없음.

제 3 절 로지스틱 모형을 이용한 분류

- 주어진 입력변수 x 에 대하여 로지스틱 함수를 이용하여 출력변수 Y 가 1 이 될 확률 $\mathbb{P}(Y = 1|x)$ 를 추정
- 0 과 1 사이의 적당한 수 c 를 절단값(cut-off value) 로 선택
- $\mathbb{P}(Y = 1|x)$ 가 c 보다 크면 자료를 $Y = 1$ 인 그룹에 할당하고 $\mathbb{P}(Y = 1|x)$ 가 c 보다 작으면 자료를 $Y = 0$ 인 그룹에 할당

절단값의 선택

절단값 c의 결정은 여러 가지를 고려하여 결정

- 고려사항 1: 사전정보
  사전정보에 의하여 두번째 범주의 자료($y = 1$ 인 자료)가 많다면, 절단값을 작은 값으로 정함
- 고려사항 2: 손실함수
  두 번째 범주의 자료를 잘못 분류하는 손실이 첫 번째 범주의 자료를 잘못 분류하는 것에 비하여 손실 정도가 심각하게 큰 경우에는 절단값 c를 작게 잡음
- 그 외 고려사항: 전문가 의견, 민감도와 특이도 등

로지스틱 모형을 이용한 분류 경계

- 로지스틱 모형의 분류 경계는 선형
  $\mathbb{P}(Y = 1|x) > c$ 이면 그룹 1에 분류 $\iff$ $\log(\mathbb{P}(Y = 1|x)/(1 - \mathbb{P}(Y = 1|x)) > \log(1 + c)/c$ 이면 그룹 1로 분류
  즉, $a + bx > log(c*)$ 이면 그룹 1에 분류
- 분류경계가 비선형인 경우 다항 로지스틱 모형으로 추정할 수 있음
- 의사결정나무나 신경망모형 등은 비선형 분류 경계를 찾는 방법들임

제 4 절 로지스틱 모형의 특징

- 선형회귀모형과의 유사성으로 인하여 사용이 쉬움
- 회귀계수와 오즈비를 이용하여 해석이 편리함
- 불필요한 변수를 제거하고 꼭 필요한 변수만을 골라냄으로써 모형의 예측력과 해석력을 높일 수 있음
- 로지스틱 모형의 변수간의 관계는 기본적으로 선형모형으로 비선형 모형을 고려하기 위하여는 다항 로지스틱회귀모형이나 의사결정나무, 신경망 모형 등의 비모수적 방법을 사용

제 5 절 R 예제 : Sonar 데이터

- Gorman and Sejnowski in their study of the classification of sonar signals using a neural network.
- Input variable is in the range 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period of time.
- Class: "R" if the object is a *rock* and "M" if it is a *mine (metal cylinder)*.
- A data frame with 208 observations on 61 variables, all numerical and one (the Class) nominal.

```
> library(mlbench)
> data(Sonar)
> t.idx = sample(1:208, 104)
> Sonar.tr = Sonar[t.idx,]
> Sonar.te = Sonar[-t.idx,]
> logit1 = glm(Class~., Sonar.tr, family=binomial())
Warning messages:
1: algorithm did not converge in: glm.fit(x = ..
2: fitted probabilities numerically 0 or 1 occurred in:
    glm.fit(x = X, y = Y, ...

> logit1
Call:  glm(formula = Class ~ ., family = binomial(), data = Sonar.tr)
Coefficients:
(Intercept)           V1           V2           V3
   143.3421    -336.9494    -195.2610     803.1947
         V4           V5           V6           V7
  -363.9940      37.2100    -138.6399     115.5241
         V8           V9          V10          V11
    -7.0440    -129.1505      19.7847      46.0523
```

```
        V12           V13           V14           V15
     0.5806    -175.1533      46.4356      -7.6232
     ...
Degrees of Freedom: 103 Total (i.e. Null);  43 Residual
Null Deviance:      144.1
Residual Deviance: 1.861e-09    AIC: 122

> prob = predict(logit1, Sonar.te, type="response")
> p.class = ifelse(prob>0.5, "R", "M")
> table(Class=Sonar.te$Class, pred=p.class)
     pred
Class  M  R
    M 32 28
    R 11 33
> mean(Sonar.te$Class != p.class)
[1] 0.375
```

제 4 장

의사결정나무

제 1 절 의사결정나무에 대한 소개

- 지도학습 기법으로 각 변수의 영역을 반복적으로 분할함으로써 전체 영역에서의 규칙 생성
- 적용결과에 의해 if-then 으로 표현되는 규칙이 생성
- 규칙의 이해가 쉽고 SQL과 같은 DB 언어로 표현하기 쉬움
- 해석이 쉬움

예측력과 해석력

- 예측력만이 중요한 경우
  예: 홍보책자 발송회사가 기대집단의 사람들이 가장 많은 반응을 보일 고객 유치 방안을 위한 예측
- 많은 경우 결정을 내리게 된데 대한 이유를 설명하는 것(해석력)이 중요
  예: 은행의 대출심사 결과 부적격 판정이 나온 경우 고객에게 부적격 이유를 설명하여야 함
- 의사결정나무는 해석력이 좋음

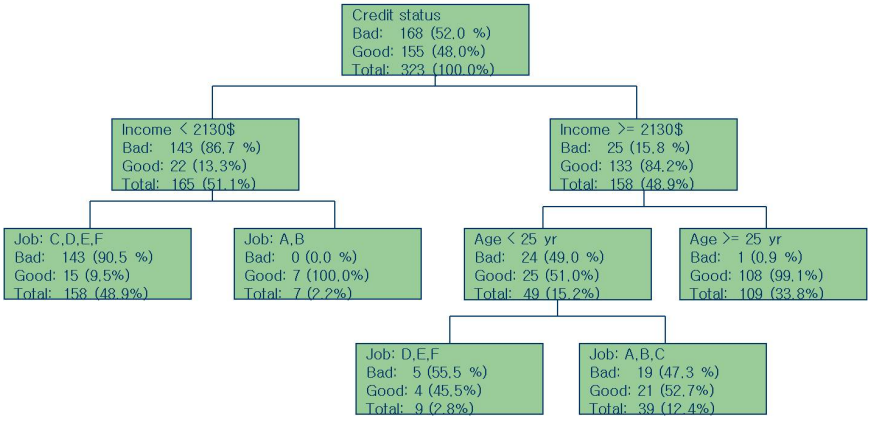


그림 4.1: 의사결정나무에 대한 예

의사결정나무의 구성요소

- 뿌리마디 (root node): 시작되는 마디로 전체 자료로 구성
- 자식마디 (child node): 하나의 마디로부터 분리되어 나간 2 개 이상의 마디들
- 부모마디 (parent node): 주어진 마디의 상위마디
- 끝마디 (terminal node): 자식마디가 없는 마디
- 중간마디 (internal node): 부모마디와 자식마디가 모두 있는 마디
- 가지 (branch): 뿌리마디로부터 끝마디까지 연결된 마디들
- 깊이 (depth): 뿌리마디부터 끝마디까지의 중간마디의 수

의사결정나무 구축을 위한 질문

앞에서 소개된 의사결정나무를 보면 다음과 같은 질문을 던질 수 있다.

- 뿌리마디의 질문이 왜 소득인가?
  : 분할기준 (splitting rule) 의 선택
- 4 번, 5 번, 7 번 마디들은 끝마디인 반면 6 번 마디는 왜 중간마디인가?
  : 분할을 계속할 것인가 그만둘 것인가 (stopping and pruning rule)

- 7번 마디에 속하는 자료는 신용상태를 어떻게 결정하여야 하는가?
  : 각 끝마디에서의 예측값 할당

제 2 절 의사결정나무의 형성과정

- 나무의 성장(growing): 각 마디에서 적절한 최적의 분리규칙을 찾아서 나무를 성장 시킴. 정지규칙을 만족하면 중단.
- 가지치기(pruning): 오분류율을 크게 할 위험이 높거나 부적절한 추론규칙을 가지고 있는 가지를 제거. 또한, 불필요한 가지를 제거.
- 타당성 평가: 이익도표(gain chart)나 위험도표(risk chart) 또는 테스트 자료 (test sample)를 사용하여 의사결정나무를 평가
- 해석 및 예측: 구축된 나무모형을 해석하고 예측모형을 설정

분리규칙

- 각 마디에서 분리규칙은 분리에 사용될 입력변수 (분리변수, split variable)의 선택과 분리가 이루어 질 기준 (분리 기준, split criterion)을 의미
- 분리에 사용될 변수(X)가 연속 변수인 경우에는 X가 분리기준 c보다 작으면 왼쪽 자식마디로 X가 c보다 크면 오른쪽 자식마디로 자료를 분리
- 분리변수가 범주형인 경우에는 분리기준은 전체 범주를 두 개의 부분집합으로 나누는 것이 됨. 예를 들면, 전체 범주가 {1,2,3,4}일때 분리기준의 예로는 {1,2,4}과 {3}이 되고 분리변수가 범주 {1,2,4}에 속하면 왼쪽자식마디로 범주 {3}에 속하면 오른쪽 자식마디로 자료를 분리

순수도와 불순도

- 각 마디에서 분리변수와 분리기준은 목표변수의 분포를 가장 잘 구별해주도록 정함

- 목표변수의 분포를 얼마나 잘 구별하는가에 대한 측정치로 순수도 (purity) 또는 불순도 (impurity)를 사용
  (예) 그룹 0과 그룹 1의 비율이 45%와 55%인 마디는 각 그룹의 비율이 90%와 10%인 마디에 비하여 순수도가 낮다 또는 불순도가 높다라고 함

- 각 마디에서 분리변수와 분리 기준의 설정은 생성된 두 개의 자식마디의 순수도의 합이 가장 큰 (혹은 불순도의 합이 가장 작은) 분리변수와 분리기준을 선택

불순도의 측도

- 분류모형 (범주형 목표변수)
  - 카이제곱 통계량 (chi-square statistics)
  - 지니지수 (Gini index)
  - 엔트로피지수 (Entropy index)
- 회귀모형 (연속형 목표변수)
  - 분산분석에 의한 F- 통계량 (F-Statistics)
  - 분산의 감소량

카이제곱 통계량

주어진 분리변수와 분리기준에 의하여 다음의 표가 주어졌다.

	Good	Bad	Total
left	32	48	80
right	178	42	220
Total	210	90	300

위의 표를 실제도수(O)라고 한다. 앞의 표에서 각 셀에 대한 기대도수(E)를 다음과 같이 구할 수 있다.

	good	bad	total
left	$300\frac{80}{300}\frac{210}{300}$	$300\frac{80}{300}\frac{90}{300}$	80
left	=56	=24	80
right	154	66	220
total	210	90	300

- 카이제곱 통계량 = ((기대도수-실제도수)의 제곱 / 기대도수)의 합
  (예) 앞의 표에서 카이제곱통계량은

$$\frac{(56-32)^2}{56}+\frac{(24-48)^2}{24}+\frac{(154-178)^2}{154}+\frac{(66-42)^2}{66}=46.75$$

- 카이제곱통계량이 최대가 되는 분리변수와 분리기준을 사용

지니지수

- 지니지수는 다음과 같이 정의된다.

$$\begin{aligned}\text{지니지수} &= 2\mathbb{P}(\text{left 에서 good})\mathbb{P}(\text{left 에서 bad})\mathbb{P}(\text{left})\\ &\quad +2\mathbb{P}(\text{right 에서 good})\mathbb{P}(\text{right 에서 bad})\mathbb{P}(\text{right})\end{aligned}$$

  (예) 앞의 표에서 지니지수를 구하면

$$2\frac{32}{80}\frac{48}{80}\frac{80}{300}+2\frac{178}{220}\frac{42}{220}\frac{220}{300}=0.355$$

- 모든 분리변수와 분리기준에서 지니지수를 가장 작게 하는 분리변수와 분리기준 선택

엔트로피 지수

- 엔트로피는 다음과 같이 정의된다.

$$\begin{aligned}\text{엔트로피}(\text{left}) &= -\mathbb{P}(\text{left 에서 good})\log_2\mathbb{P}(\text{left 에서 good})\\ &\quad -\mathbb{P}(\text{left 에서 bad})\log_2\mathbb{P}(\text{left 에서 bad})\end{aligned}$$

- 엔트로피 지수 = 엔트로피(left)$\mathbb{P}$(left) + 엔트로피(right)$\mathbb{P}$(right)

  (예) 앞의 표에서 엔트로피 지수를 구하면

$$-\left(\frac{32}{80}\log_2\left(\frac{32}{80}\right)+\frac{48}{80}\log_2\left(\frac{48}{80}\right)\right)\frac{80}{300}$$
$$-\left(\frac{178}{220}\log_2\left(\frac{178}{220}\right)+\frac{42}{220}\log_2\left(\frac{42}{220}\right)\right)\frac{220}{300}=.7747$$

분리방법 예제

아래의 자료에 대하여 지니지수를 이용하여 최적의 분리를 찾아보자.

Temperature	Humidity	Windy	Class
Hot	High	False	N
Hot	High	True	N
Hot	High	False	P
Mild	High	False	P
Cold	Normal	False	P
Cold	Normal	True	N
Cold	Normal	True	P
Mild	High	False	N
Cold	Normal	False	N
Mild	Normal	False	P
Mild	Normal	True	P
Mild	High	True	P
Hot	Normal	False	N
Mild	High	True	P

Temperature를 기준으로 분리

- left={Hot}, right = {Mild,Cold}

	N	P	total
left	3	1	4
right	3	7	10
total	6	8	14

Gini index=$2\frac{1}{4}\frac{3}{4}\frac{4}{14} + 2\frac{3}{10}\frac{7}{10}\frac{10}{14} = 0.4071$

- left={Mild}, right = {Hot,Cold}

	N	P	total
left	1	5	6
right	5	3	8
total	6	8	14

Gini index=$2\frac{1}{6}\frac{5}{6}\frac{6}{14} + 2\frac{5}{8}\frac{3}{8}\frac{8}{14} = 0.3869$

- left={Cold}, right = {Hot,Mild}

	N	P	total
left	2	2	4
right	4	6	10
total	6	8	14

Gini index=$2\frac{2}{4}\frac{2}{4}\frac{4}{14} + 2\frac{5}{10}\frac{6}{10}\frac{10}{14} = 0.4860$

Humidity를 기준으로 분리

- left={High}, right = {Normal}

	N	P	total
left	3	4	7
right	3	4	7
total	6	8	14

- Gini index=$2\frac{3}{7}\frac{4}{7}\frac{7}{14} + 2\frac{3}{7}\frac{4}{7}\frac{7}{14} = 0.4897$

Windy를 기준으로 분리

- left={False}, right = {True}

	N	P	total
left	4	4	8
right	2	4	6
total	6	8	14

- Gini index=$2\frac{4}{6}\frac{2}{6}\frac{6}{14} + 2\frac{4}{8}\frac{4}{8}\frac{8}{14} = 0.4762$

결과를 종합하여 불순도가 가장 작은 분리를 선택하므로 Temperature에 대하여 left = {Mild}, right={Hot, Cold}로 분리하는 것이 가장 좋음

회귀모형에서 불순도의 측정

- 오른쪽 자식마디와 왼쪽 자식마디의 평균의 차이를 검정하는 t- 통계량의 유의확률이 가장 작은 분리변수와 분리기준을 사용하여 분리
- 왼쪽 자식마디의 자료의 분산과 오른쪽 자식마디의 자료의 분산의 합이 가장 작은 분리를 선택

불순도에 대한 참고사항

- 불순도는 각 마디마다 정의됨
- 불순도를 이용한 분리기준의 선택에서는 자식마디의 불순도의 합을 가장 작게 하는 분리를 선택
- 이 방법은 부모마디의 불순도에서 자식마디의 불순도의 합의 차이가 최대가 되는 분리를 찾는 것과 동일함
- 여러 개의 마디 중에서 최적의 분리를 찾는 경우에는 자식마디의 불순도의 합을 사용하지 않고, 부모마디와 자식마디 사이의 불순도의 차이를 최대로 하는 분리를 찾음

정지규칙

- 현재의 마디가 더 이상 분리가 일어나지 못하게 하는 규칙
- 규칙의 종류
  - 모든 자료가 한 그룹에 속할 때
  - 마디에 속하는 자료가 일정 수 이하일 때
  - 불순도의 감소량이 아주 작을 때
  - 뿌리마디로부터의 깊이가 일정 수 이상일 때 등이 있음

가지치기

- 지나치게 많은 마디를 가지는 (복잡한 모형) 의사결정나무는 새로운 자료에 적용할 때 예측오차가 매우 클 가능성이 있음
- 성장이 끝난 나무의 가지를 제거하여 적당한 크기를 갖는 나무모형을 최종적인 예측모형으로 선택하는 것이 예측력의 향상에 도움이 됨
- 적당한 크기를 결정하는 방법은 평가용 자료(validation data)를 사용하여 예측에러를 구하고 이 예측에러가 가장 작은 나무모형을 선택

제 3 절 여러가지 알고리즘

CART

- Classification And Regression Tree
- 1984년 Breiman과 그의 동료들이 발명
- 기계학습(machine learning) 실험의 산물
- 가장 널리 사용되는 의사결정나무 알고리즘
- 이지분류(binary split)를 이용

- 불순도: 목표변수가 범주형인 경우 지니지수를 이용하고 목표변수가 연속형인 경우에는 분산을 이용
- CART 에서는 압력변수들의 선형결합 중에서 최적의 분리를 찾기도 함

C4.5

- 호주의 연구원 J. Ross Quinlan 에 의하여 개발
- 초기버전은 ID 3 (Iterative Dichotomizer 3) 로 1986 년에 개발
- CART 와는 다르게 각 마디에서 다지분리 (multiple split) 가 가능
- 범주형 입력변수에 대해서는 범주의 수만큼 분리가 일어남
- 불순도: 엔트로피 지수를 사용

CHAID

- Chi-squared Automatic Interaction Detection
- 1975 년 J.A. Hartigan 이 발표
- 1963 년 J.A. Morgan 과 N.A. Souquist 에 의해 서술된 AID 의 후신
- 가지치기를 하지 않고 나무를 적당한 크기에서 성장을 중지
- 입력변수가 반드시 범주형
- 불순도: 카이제곱 통계량을 사용

제 4 절 의사결정나무의 특징

의사결정나무의 장점

- 이해하기 쉬운 규칙을 생성
- 분류작업이 용이

- 연속형 변수와 범주형 변수를 모두 다 취급할 수 있음
- 가장 좋은 변수를 명확히 알아낼 수 있음
- 이상치에 덜 민감
- 모형의 가정 (선형성, 등분산성 등)이 필요 없는 비모수적 모형

의사결정나무의 단점

- 목표변수가 연속형인 회귀모형에서는 그 예측력이 떨어짐
- 나무가 너무 깊은 경우에는 예측력의 저하뿐 아니라 해석도 어려움
- 계산량이 많을 수 있음
- 분류경계가 비사각인 경우 문제가 있음
- 결과가 불안정

제 5 절 R 예제

Iris Data

```
> library(MASS)
> library(tree)
> data(iris)
> ir.tr = tree(Species ~., iris)
> summary(ir.tr)

Classification tree:
tree(formula = Species ~ ., data = iris)
Variables actually used in tree construction:
[1] "Petal.Length" "Petal.Width"  "Sepal.Length"
Number of terminal nodes:  6
```

```
Residual mean deviance:  0.1253 = 18.05 / 144
Misclassification error rate: 0.02667 = 4 / 150
> ir.tr
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 150 329.600 setosa ( 0.33333 0.33333 0.33333 )
   2) Petal.Length < 2.45 50   0.000 setosa ( 1.00000 0.00000 0.00000 ) *
   3) Petal.Length > 2.45 100 138.600 versicolor ( 0.00000 0.50000 0.50000 )
     6) Petal.Width < 1.75 54  33.320 versicolor ( 0.00000 0.90741 0.09259 )
      12) Petal.Length < 4.95 48   9.721 versicolor ( 0.00000 0.97917 0.02083 )
        24) Sepal.Length < 5.15 5   5.004 versicolor ( 0.00000 0.80000 0.20000 ) *
        25) Sepal.Length > 5.15 43   0.000 versicolor ( 0.00000 1.00000 0.00000 ) *
      13) Petal.Length > 4.95 6   7.638 virginica ( 0.00000 0.33333 0.66667 ) *
     7) Petal.Width > 1.75 46   9.635 virginica ( 0.00000 0.02174 0.97826 )
      14) Petal.Length < 4.95 6   5.407 virginica ( 0.00000 0.16667 0.83333 ) *
      15) Petal.Length > 4.95 40   0.000 virginica ( 0.00000 0.00000 1.00000 ) *
```

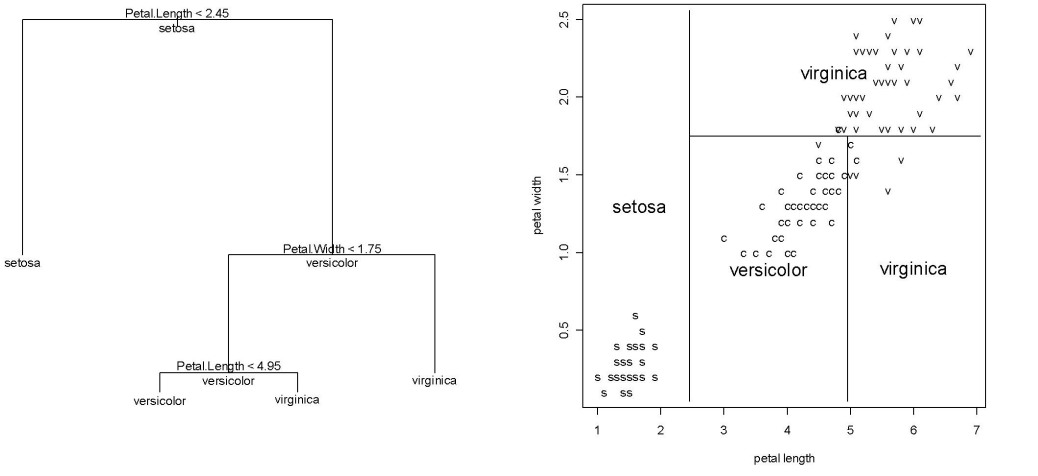


그림 4.2: iris 데이터에 대한 분류 트리와 분할

```
> ir.tr1 = snip.tree(ir.tr, nodes = c(12, 7))
```

```
> plot(ir.tr1)
> text(ir.tr1, all = T)
> par(pty = "s")
> plot(iris[, 3],iris[, 4], type="n",
     xlab="petal length", ylab="petal width")
> text(iris[, 3], iris[, 4], c("s", "c", "v")[iris[, 5]])
> partition.tree(ir.tr1, add = TRUE, cex = 1.5)
```

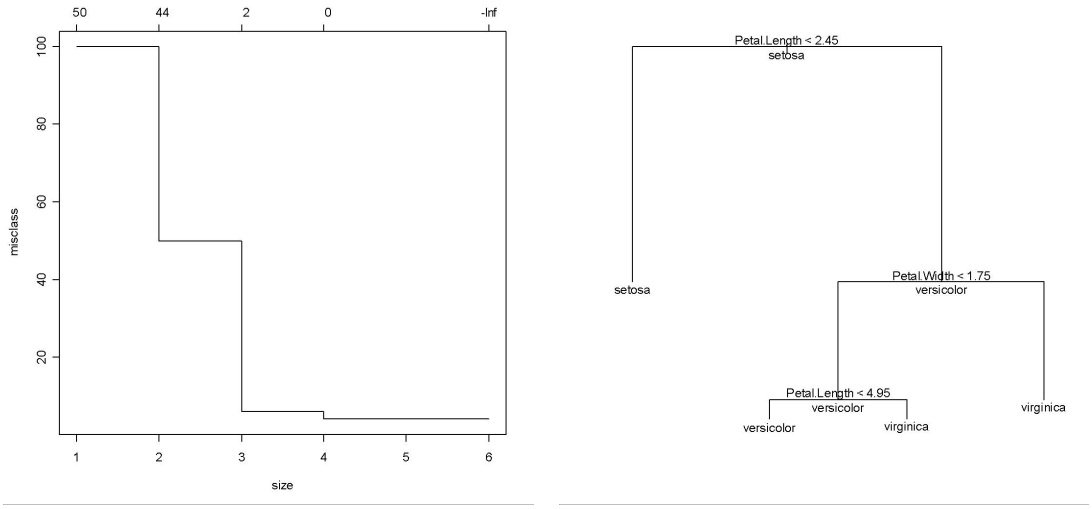


그림 4.3: Iris 데이터에 대한 오분류율을 이용한 가지치기과 가지치기된 트리

```
> plot(prune.misclass(ir.tr))
> fin.tr = prune.misclass(ir.tr, best=4)
> plot(fin.tr)
> text(fin.tr, all = T)
```

제 5 장

신경망

제 1 절 신경망에 대한 소개

신경망이란 ?

- 인간의 두뇌구조를 모방한 지도학습 방법
- 여러 개의 뉴런들이 상호 연결하여 입력값에 대한 최적의 출력값을 예측
- 장점 : 좋은 예측력
- 단점 : 해석이 어려움

신경망의 배경

- 1940 년대 McCulloch 와 Pits 에 의해 인간 뇌의 신경노드의 작동 모형 구축
- 1950 년대 Rosenblatt 에 의하여 지도학습에 응용될 수 있는 단층 신경망 Perceptron 개발
- 1980 년대 이전에는 컴퓨터 성능이 낮아서 그리 널리 쓰이지 않음
- 1980 년대에 Hopfield 에 의해 다시 각광을 받기 시작함
- 다층 신경망 (multi-layer perceptron) 과 역전파 (back propagation) 알고리즘의 결합은 신경망 모형의 응용분야를 크게 넓힘

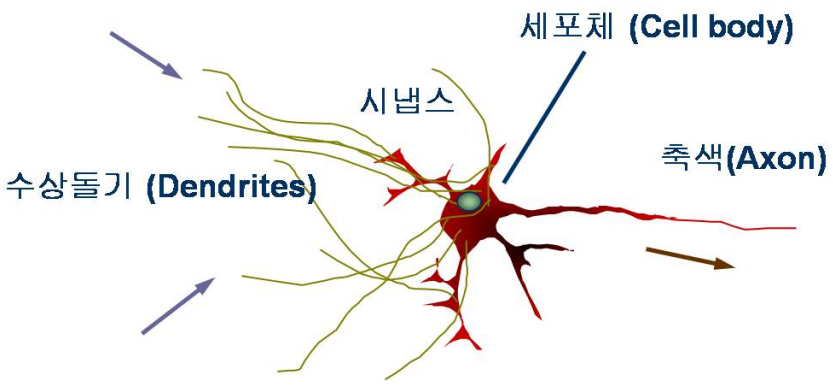


그림 5.1: 인간의 신경세포

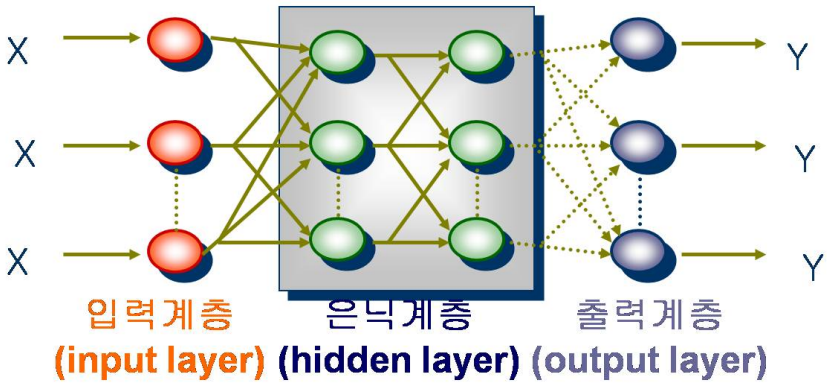


그림 5.2: 신경망 모형의 구조

용어 정리

- MLP (multi-layer perceptron): 입력층, 은닉층, 출력층으로 구성
- SLP (single-layer perceptron): 입력층과 출력층으로만 구성
- 입력층: 각 입력변수에 대응되는 노드로 구성. 노드의 수는 입력변수의 개수와 같음
- 은닉층: 입력층으로부터 전달되는 변수값들의 선형결합을 비선형함수로 처리하여 출력층 또는 다른 은닉층에 전달
- 출력층: 목표변수에 대응되는 노드. 분류모형에서는 그룹의 수 만큼의 출력노드가 생성

제 2 절 다층 신경망 (MLP)

- 각함수구조

$$\text{Output} = f_2(w_4 + w_5 z), \quad z = f_1(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3)$$

- 함수 f_1 과 f_2 를 활성함수라고 함
- 회귀모형 (출력변수가 연속형인 경우에는) 는 $f_2(x) = x$ 가 많이 쓰임
- 분류모형인 경우에는 f_2 에 시그모이드 (sigmoid) 함수가 많이 사용됨
- f_1 에는 시그모이드 함수가 사용됨

시그모이드 함수

- 시그모이드 함수는 단극성, 양극성 두 종류가 있음
- 단극성 시그모이드 함수:
  - $f(x) = \frac{1}{1+e^{-x}}$

  - 증가함수이며 출력값이 0과 1사이의 값을 갖음
  - 로지스틱함수와 유사

- 양극성 시그모이드 함수
  - $f(x) = \frac{1-e^{-x}}{1+e^{-x}}$
  - 증가함수이며 출력값이 -1과 1사이의 값을 갖으며 $f(0) = 0$

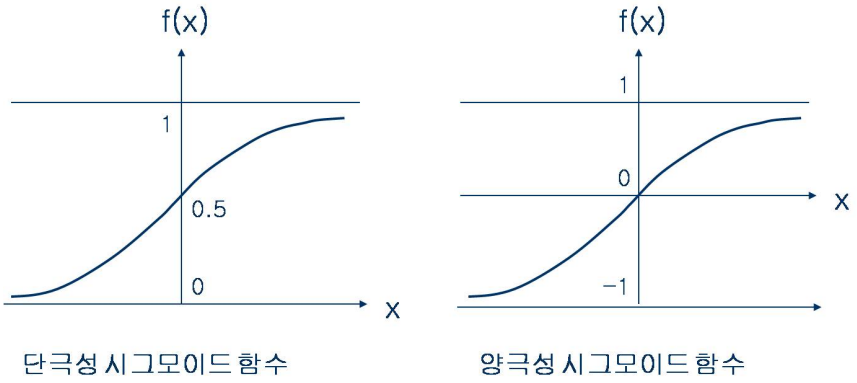


그림 5.3: 시그모이드 함수 그림

다양한 MLP

- 은닉노드의 수가 k개 인 일반적인 MLP의 함수구조는 다음과 같다.

$$\text{Output} = f_2(w_{20} + w_{21}z_1 + \cdots + w_{2k}z_k,$$

$$z_i = f_1(w_{1i0} + w_{1i1}x_1 + \cdots + w_{1ip}x_p), i = 1, \ldots, k$$

- 활성함수, 특히, 은닉노드의 활성함수를 바꿈으로써 여러 가지 신경망 모형을 만들 수 있음
  (예) RBF (Radial Basis function) 신경망

목적함수

- 신경망 모형을 구축하는 방법은 크게 두 단계로 이루어 짐

- 1 단계: 적절한 수의 은닉층과 은닉노드의 수 결정
- 2 단계: 연결강도 (weight) 를 추정

- 연결강도의 추정은 주어진 목적함수를 최소화 (또는 최대화) 를 통하여 추정
- 일반적으로 많이 쓰이는 목적함수로는 선형모형에 쓰이는 오차제곱합 $\sum(y_i - p_i)^2$ 이다. 여기서 y_i 는 i 번째 패턴의 실제 값이고 p_i 는 i 번째 패턴의 예측값
- 분류 모형에서 사용되는 또 다른 목적함수로는 로그우도함수 (log likelihood) 를 사용. 목적변수가 이진형인 경우

$$-\sum \left(y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)\right)$$

역전파 (Back propagation) 알고리즘

- 신경망의 목적함수는 연결강도에 대하여 비선형 함수인데 이의 최적화를 위해 역전파 방법이 널리 사용됨
- 역전파 알고리즘
  - 1 단계: 주어진 연결강도을 이용하여 예측값을 계산
  - 2 단계: 실제 출력값과 예측값 사이의 오차를 계산
  - 3 단계: 오차를 은닉층과 입력층으로 역전파 시켜서 연결강도를 새로 조절

  위의 3 단계를 초기 연결강도를 이용하여 계속 반복. 연결강도의 값이 일정하게 유지되면 반복을 멈춤
- 역전파시 학습률 (learning rate) 가 필요한데, 이 값은 처음에는 크게, 그리고 반복수가 증가 하면서 점점 작아지도록 정하는 것이 좋음

신경망 구축 4 단계

- 자료의 선택 및 적절한 변환
- 신경망모형의 은닉층의 노드 수와 활성함수의 결정 (신경망 모형 선택)

- 연결강도의 최적화
- 결과의 해석

불순도에 대한 참고사항

- 불순도는 각 마디마다 정의됨
- 불순도를 이용한 분리기준의 선택에서는 자식마디의 불순도의 합을 가장 작게 하는 분리를 선택
- 이 방법은 부모마디의 불순도에서 자식마디의 불순도의 합의 차이가 최대가 되는 분리를 찾는 것과 동일함
- 여러 개의 마디 중에서 최적의 분리를 찾는 경우에는 자식마디의 불순도의 합을 사용하지 않고, 부모마디와 자식마디 사이의 불순도의 차이를 최대로 하는 분리를 찾음

신경망모형 선택 방법

- Trial and error
  - 은닉층 수, 은닉층의 노드 수, 활성함수를 변화시켜 가면서 가장 좋은 모형을 찾음
  - 비용과 시간이 많이 소요됨
  - 실제 분석에서는 가장 많이 쓰이는 방법
- Constructive algorithm
  - 작은 모형에서 시작해 큰 모형으로 키워나감
  - 은닉층의 노드의 수를 결정 할 때 많이 사용
  - 다른 수의 은닉 노드를 갖는 신경망들을 비교할 수 있는 통계량이 필요
  - 실제 분석에서는 노드 수를 증가시키며 학습오차의 감소량을 측정하고 이 감소량이 급격히 줄어드는 곳의 노드 수를 최종 모형으로 선택

- Pruning
  - 큰 모형에서 시작하여 작은 모형으로 축소시키는 방법
  - 의사결정나무의 pruning(가지치기) 방법과 유사
  - 일반적으로, pruning (후진소거법)이 constructive algorithm (전진선택법)에 비해 좋은 결과를 제공
  - 하지만, pruning방법은 큰 모형을 결정하는 문제가 쉽지 않으며, 큰 모형에는 많은 수의 모수가 있어서 그 추정이 쉽지 않음
- Regularization
  - 모형의 복잡도의 크기에 비례하는 penalty를 넣은 적합도 함수를 최소화
  - 선형모형에서 사용되는 Mallow's Cp나 AIC의 사용과 유사
  - penalty를 결정하기가 쉽지 않음
  - penalty에 따라 최종 결과가 많은 차이를 보임

제 3 절 신경망 구축시 고려사항

입력자료 선택

- 신경망모형은 복잡성으로 인하여 입력자료의 선택에 매우 민감
- 좋은 입력자료란
  - 범주형 입력변수가 모든 범주에서 일정 빈도 이상의 값을 갖음
  - 연속형 입력변수 값들의 범위가 변수 간에 많은 차이가 없음
  - 입력변수의 수가 너무 많지 않음 (물론, 너무 적어도 안됨)
  - 범주형 출력값의 각 범주의 빈도가 비슷함

연속형 변수의 변환 / 범주화

- 연속형변수의 경우 그 분포가 평균을 중심으로 대칭이 아닌 경우에는 안 좋은 결과를 생성
  (예) 소득 : 일반적으로 대부분의 고객의 소득은 평균 미만이고 아주 특정한 고객의 소득이 매우 큰 패턴을 보임

- 분포가 대략 대칭이 되도록 변환 (예: log 변환)

- 다른 방법으로는 연속변수의 범주화
  (예) 소득을 매우 낮음, 낮음, 중간, 높음, 대단히 높음 등으로 범주화. 각 범주의 빈도가 비슷하게 되도록 설정하는 것이 바람직

새로운 범주의 생성

- 때로 원 자료의 변수들을 조합하여 새로운 변수를 만든 후 이 변수를 입력변수로 사용하면 아주 좋은 결과를 얻을 수 있음
  (예) 고객의 여러 가지 사항을 고려한 (수입, 학력 등) 구매지수를 만든 후 이 지수를 입력으로 하여 특정한 상품의 구매여부를 예측

범주형 변수의 가변수화

- 신경망에서도 범주형 변수를 가변수화 시켜야 함

- 선형모형과는 다르게, 신경망 모형에서는 가변수화를 어떻게 하느냐에 따라서도 그 결과가 민감하게 반응
  (예) 남자와 여자를 0과 1로 만든 경우와 ?1과 1로 만든 경우의 결과가 틀릴 수 있음

- 모든 범주형 변수는 같은 크기로 가변수화 하는 것이 좋음

제 4 절 신경망 모형의 특징

- 범용 근사자 (universal approximator)

  - 어떠한 분류함수도 근사적으로 표현할 수 있음
  - 은닉층이 하나인 MLP는 범용 근사자

- 최적화와 비 수렴성
  - 역전파 알고리즘이 제공하는 모형이 최적이 아닌 경우가 많이 발생
  - 특히 역전파알고리즘의 초기값에 민감
  - 초기 값을 바꾸어 가며 여러 개의 신경망을 만든 후 그 중 최적의 모형을 선택

- 해석의 어려움
  - 입력변수와 출력변수와의 관계를 파악하는 것이 거의 불가능
  - 신경망모형은 선형모형이나 의사결정나무에 비해 예측력은 뛰어남
  - 실제 분석에서는 신경망과 의사결정나무를 같이 사용하는 방법도 고려

- 과적합
  신경망모형에서 적절한 크기의 모형을 선택하기가 어려움. 비 숙련자가 사용하기가 어렵고, 많은 경우 과적합의 위험이 있음

제 5 절 R 예제

Iris 데이터

```
plot(iris[,1:3], col=as.integer(iris$Species),
    pch=substring((iris$Species),1,1))
```

Partition data into training data and test data

```
data(iris)
# use half the iris data
samp <- c(sample(1:50,25), sample(51:100,25), sample(101:150,25))
iris.tr<-iris[samp,]
iris.te<-iris[-samp,]
```

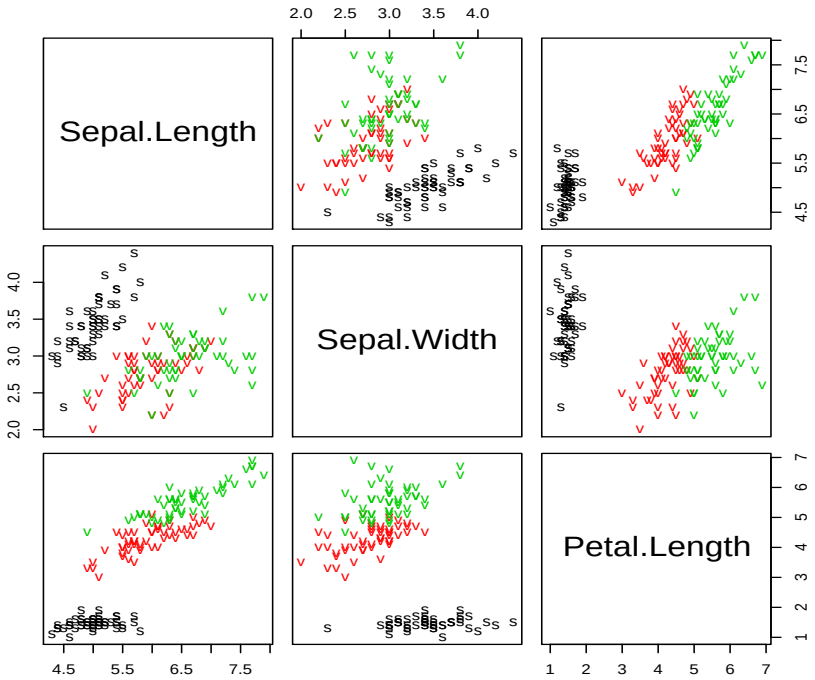


그림 5.4: Iris 자료

Neural Network modeling

```
ir1 <- nnet(Species~., data=iris.tr, size = 2, decay = 5e-4)


# weights:  19
initial  value 83.513437
iter  10 value 1.427673
iter  20 value 0.620816
iter  30 value 0.526732
iter  40 value 0.484637
iter  50 value 0.445681
...
iter  90 value 0.370717
iter 100 value 0.369270
final  value 0.369270
```

```
stopped after 100 iterations
```

View Neural network model

```
> names(ir1)
"value"     : 에러함수의 값
"wts"       : 모수추정치
"fitted.values" : output추정치
"residuals" : 잔차

> summary(ir1)
    a 4-2-3 network with 19 weights
    options were - softmax modelling  decay=5e-04
    b->h1 i1->h1 i2->h1 i3->h1 i4->h1 #은닉노드 1
    -6.48  -5.24  -3.83   8.15   6.37
    b->h2 i1->h2 i2->h2 i3->h2 i4->h2 #은닉노드 2
    0.39   0.61   1.79  -3.02  -1.30
    b->o1 h1->o1 h2->o1 #출력노드 1
    -2.48  -1.83   9.14
    b->o2 h1->o2 h2->o2 #출력노드 2
    5.96  -9.13  -7.81
    b->o3 h1->o3 h2->o3 #출력노드 3
    -3.54  10.84  -1.26
```

Model 평가: Test error

```
y<-iris.te$Species
p<- predict(ir1, iris.te, type = "class")

tt<-table(y, p)

            p
y            setosa versicolor virginica
```

```
  setosa         25          0         0
  versicolor      0         21         4
  virginica       0          0        25
```

Hidden unit의 수에 따른 Test error

```
test.err<-function(h.size)
{
    ir <- nnet(Species~., data=iris.tr, size = h.size,
        decay = 5e-4, trace=F)
    y<-iris.te$Species
    p<- predict(ir, iris.te, type = "class")
    err<-mean(y != p)
    c(h.size, err)
}

out<-t(sapply(2:10, FUN=test.err))
pdf("nntest.pdf")
plot(out, type="b", xlab="The number of Hidden units",
    ylab="Test Error")
dev.off()
```

제 6 장

연관성분석

제 1 절 연관성 분석에 대한 소개

- 데이터 안에 존재하는 항목간의 연관규칙 (association rule) 을 발견하는 과정
- 연관규칙: 상품을 구매하거나 서비스를 받는 등의 일련의 거래나 사건들의 연관성에 대한 규칙
- 연관성 분석을 마케팅에서 손님의 장바구니에 들어있는 품목간의 관계를 알아본다는 의미에서 장바구니분석 (market basket analysis) 이라고도 함

왜 연관성 분석을 하는가 ?

- 고객의 슈퍼마켓에서 구입한 물건들이 담겨져 있는 장바구니의 정보를 생각해보자.
- 연관성 분석은, 특정한 상품을 구입한 고객이 어떤 부류에 속하는지, 그들이 왜 그런 구매를 했는지 알기 위해서 고객들이 구매한 상품에 대한 자료를 분석하는 것
- 이러한 분석을 통하여 효율적인 매장진열, 패키지 상품의 개발, 교차판매전략 구사, 기획상품의 결정 등에 응용할 수 있음

연관성 분석의 응용

- 백화점이나 호텔에서 고객들이 다음에 원하는 서비스를 미리 알 수 있음

- 신용카드, 대출 등의 은행서비스 내역으로부터 특정한 서비스를 받을 가능성이 높은 고객의 탐지 가능
- 입력층: 각 입력변수에 대응되는 노드로 구성. 노드의 수는 입력변수의 개수와 같음
- 의료보험금이나 상해보험금 청구에서 특이한 형태를 보이는 경우 보험사기의 징조가 될 수 있고 추가적인 조사 필요
- 환자의 의무기록에서 여러 치료가 같이 이루어진 경우 합병증 발생의 징후 탐지

제 2 절 연관성 규칙

- 유용한 규칙
  (예) 목요일 식료품 가게를 찾는 고객은 아기 기저귀와 맥주를 함께 구입하는 경향이 있다
- 자명한 규칙
  (예) 한 회사의 전자제품을 구매하던 고객은 전자제품을 살 때 같은 회사의 제품을 사는 경향이 있다
- 설명이 불가능한 규칙
  (예) 새로 연 건축 자재점에서는 변기덮게가 많이 팔린다

식료품 거래내역 예제

다음의 표는 5개 제품을 취급하는 편의점에서 5번의 거래 내역이다.

고객번호	품목
1	오렌지 쥬스, 사이다
2	우유, 오렌지 쥬스, 식기세척제
3	오렌지 쥬스, 세제
4	오렌지 쥬스, 세제, 사이다
5	식기 세척제, 사이다

	오렌지 쥬스	식기세척제	우유	사이다	세제
오렌지 쥬스	4	1	1	2	2
식기세척제	1	2	1	1	0
우유	1	1	1	0	0
사이다	2	1	0	3	1
세제	2	0	0	1	2

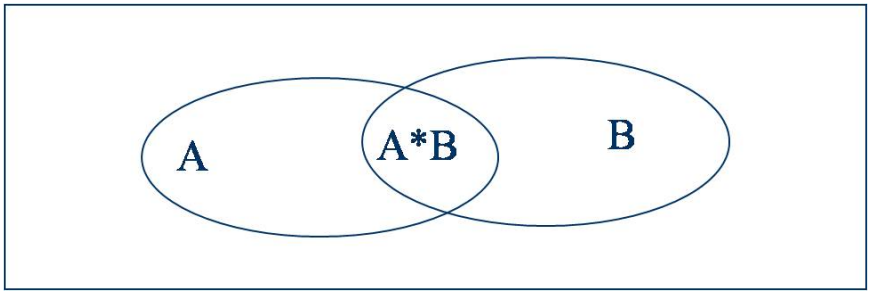
연관성 규칙의 조건

- 동시구매표로부터 간단한 규칙 (예: 사이다를 구입하는 고객은 오렌지 쥬스를 산다)을 만들 수 있음
- 연관 규칙은 “f A, then B”와 같은 형식으로 표현
- 모든 “if-then” 규칙이 유용한 규칙은 아님
- 찾은 규칙이 유용하게 사용되기 위하여는
  - 두 품목 (품목 A와 품목 B) 이 함께 구매한 경우의 수가 일정 수준 이상이여야 하며 (일정 이상의 지지도)
  - 품목 A를 포함하는 거래 중 품목 B를 구입하는 경우의 수가 일정수준 이상이여야 함 (일정 이상의 신뢰도)

제 3 절 연관성분석의 측도

지지도와 신뢰도

- 지지도(support)
  - 두 품목 A와 B의 지지도는 전체 거래항목 중 항목 A와 항목 B가 동시에 포함하는 거래의 비율
  - 지지도 $= \mathbb{P}(A \cap B)$ = A와 B가 동시에 포함된 거래수 / 전체 거래수



- 신뢰도 (confidence)

  연관성 규칙 "If A, then B" 의 신뢰도는

$$\frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)} = \frac{\text{품목 A와 B를 동시에 포함하는 거래 수}}{\text{품목 A를 포함하는 거래 수}}$$

식료품 거래내역 예제

- 연관성 규칙 "오렌지 쥬스를 사면 사이다를 구매한다" 의 지지도와 신뢰도
  - 지지도= 2/5
  - 신뢰도= 2/4
- 연관성 규칙 "우유와 오렌지 쥬스를 사면 식기세척제를 산다" 의 지지도와 신뢰도
  - 지지도= 1/5
  - 신뢰도=1/1

피자 토핑 예제

다음은 세 피자 토핑 A,B,C 의 동시거래 내역이다.

항목	거래수
A	100
B	150
C	200
A+B	400
A+C	300
B+C	200
A+B+C	100
추가안함	550

전체 거래수 = 2000

각 품목의 조합에 대한 지지도

항목	품목이 포함된 거래수	확률
A	900	0.450
B	850	0.425
C	800	0.400
A+B	500	0.250
A+C	400	0.200
B+C	300	0.150
A+B+C	100	0.050

모든 연관성 규칙에 대한 신뢰도

규칙 $X \Rightarrow Y$	$\mathbb{P}(X \cap Y)$	$\mathbb{P}(X)$	신뢰도 $\mathbb{P}(Y\|X)$
$A \Rightarrow B$	0.250	0.450	0.556
$B \Rightarrow A$	0.250	0.425	0.588
$C \Rightarrow B$	0.150	0.400	0.375
$B \Rightarrow C$	0.150	0.425	0.353
$A \Rightarrow C$	0.200	0.450	0.444
$C \Rightarrow A$	0.200	0.400	0.500
$A + B \Rightarrow B$	0.050	0.250	0.200
$B + C \Rightarrow A$	0.050	0.150	0.333
$A + C \Rightarrow B$	0.050	0.200	0.250

향상도

- 3가지 품목을 포함하는 연관성 규칙 중 가장 신뢰도가 높은 규칙은 "B와 C를 구입하면 A도 구매한다"이며 이 연관성 규칙의 신뢰도는 0.333
- 전체 거래에서 품목 B+C의 거래가 일어날 가능성은 0.15이기 때문에 이 연관성 규칙은 실질적으로 의미 있는 규칙이 못 됨
- 이 연관성 규칙은 규칙이 없는 경우보다 못한 결과를 줌
- 이러한 연관성 규칙의 성질을 향상도를 통하여 파악할 수 있음
- 정의
  - 연관성 규칙 "A이면 B이다"의 향상도는

$$= \frac{\dfrac{\text{품목 A와 B를 포함하는 거래 수} \times \text{전체 거래 수}}{\text{품목 A를 포함하는 거래 수} \times \text{품목 B를 포함하는 거래 수}}}{\mathbb{P}(A)\mathbb{P}(B)} = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)\mathbb{P}(B)} = \frac{\mathbb{P}(B|A)}{\mathbb{P}(B)} = \frac{\text{신뢰도}}{\mathbb{P}(B)}$$

  - 향상도는 품목 A가 주어지지 않았을 때의 품목 B의 확률 대비 품목 A가 주어졌을 때의 품목 B의 확률의 증가 비율임

  - 이 값이 클 수록 품목 A의 구매여부가 품목 B의 구매 여부에 큰 영향을 미침

- 해석
  - 품목 A와 품목 B의 구매가 상호 관련이 없다면 $\mathbb{P}(B|A)$와 $\mathbb{P}(B)$와 같게 되어 향상도가 1이 됨
  - 어떤 규칙의 향상도가 1보다 크면, 이 규칙은 결과를 예측하는데 있어서 우연적 기회 (random chance)보다 우수하다는 것을 의미하고
  - 향상도가 1보다 작으면 이러한 규칙은 결과를 예측하는데 있어서 우연적 기회보다 나쁨

향상도	의미
1	두 품목이 독립적인 관계
< 1	두 품목이 서로 음의 상관 관계
> 1	두 품목이 서로 양의 상관 관계

제 4 절 연관성분석의 절차

연관성 분석의 3단계

- 첫 번째 단계: 적절한 품목의 선택
- 두 번째 단계: 동시구매표를 이용하여 연관규칙을 찾아내는 단계
- 세 번째 단계: 품목의 수가 너무 많아서 생기는 문제점들의 해결

적절한 품목의 선택

- 어떤 품목을 선택할 것이냐는 문제는 전적으로 분석의 목적에 달려있음
  (예) 대형 할인점에서는 술을 하나의 상위 품목으로 고려할 수 있음. 어떤 경우에는 술을 세분화 하여 술을 소주, 양주, 맥주, 포도주, 막걸리 등 세분화
- 가상 품목
  (예) 디자이너 레벨(예: Calvin Klein), 저지방과 무지방 제품

- 탐색해야 할 규칙의 수는 고려되는 품목의 수에 따라 지수적으로 증가
- 일반적으로 일차단계에서 상위수준의 품목 분류를 이용하여 규칙을 찾은 후 이를 바탕으로 세분화된 품목으로 분석을 진행
- 연관성분석은 각 품목들의 거래회수가 비슷한 경우에 가장 효율적

연관규칙 발견

- 규칙을 어떻게 표현 하느냐가 중요
  (예) "아기 기저귀와 목요일이 주어지면 맥주가 결과"인 규칙이 "목요일이면 아기 기저귀와 맥주" 보다 유익
- 향상도가 1보다 작은 경우에는 결과를 역으로 나타내는 것이 좋음
- 시차에 따라 이루어지는 구매에서는 시차 연관성분석
  (예) 인터넷 쇼핑에서 웹페이지 방문 순서

현실적 문제의 해결

- 품목의 수가 증가하면 계산량은 기하급수적으로 증가
- 가지치기 (pruning)을 이용하여 불필요한 부분을 제거
- 최소지지도 가지치기(Minimum Support Pruning, MSP)
  : 지지도가 기준보다 작은 품목의 조합은 더 이상 품목을 추가하지 않음

제 5 절 기타 고려사항

음의 연관규칙

연관성 분석의 3단계

- 향상도가 1보다 작으면, 결과를 역으로 나타내는 것이 좋음

- 음의 연관규칙은 결과에 '이다' 대신에 '아니다'를 씀
  (예) "B와 C이면 A이다"의 신뢰도가 33%이면 "B와 C이면 A가 아니다"의 신뢰도는 67%가 됨
- 향상도가 1보다 작은 규칙의 음의 규칙은 1보다 큰 향상도를 갖음
- 그러나 음의 연관규칙은 원래의 연관규칙만큼 유용하지 않을 수 있음

시차 연관규칙

- 시계열자료와 같이 사건들이 어떤 순서로 일어날 때 이 사건들 사이의 연관성을 알아내는 것
- 흔히 같은 고객의 구매 패턴을 기반으로 구매 패턴이 시간에 따라 연관이 있는지를 알려고 할 때 사용됨

다음 자료는 환자가 X라는 병명으로 진단과 치료방법간의 시차 연관성 분석을 목적으로 한다.

환자번호	의사	순서	품목
1356	김	1	진단 X
5690	김	2	진단 X
1356	김	3	처방 2
7573	박	4	진단 X
7573	박	5	처방 2
5690	김	6	처방 1
1356	김	7	처방 1
7573	박	8	처방 2

만약 모든 환자들의 정도가 동일하고 처방 1이 처방 2보다 강도가 높다면, 의사 김은 처방 1을 과잉처방하고 있는 것임

제 6 절 연관성 분석의 특징

- 결과가 분명함 (If-then 규칙)

- 거대 자료의 분석의 시작으로 적합
- 변수의 개수가 많은 경우에 쉽게 사용될 수 있고 계산이 용이
- 품목 수의 증가에 따라 계산량이 폭증
- 연속형 변수를 사용할 수 없음
- 적절한 품목을 결정하기가 어려움
- 거래가 드문 품목에 대한 정보를 찾기가 어려움

제 7 절 R 예제

자료변환

```
library(arules)
## 1. example: <transactions자료를 리스트 형태로 변환
a_list = list(
        c("a","b","c"),
        c("a","b"),
        c("a","b","d"),
        c("c","e"),
        c("a","b","d","e")
        )
## set transaction names
names(a_list) = paste("Tr",c(1:5), sep = "")
a_list

## coerce into transactions
trans = as(a_list, "transactions")

## analyze transactions
```

```
summary(trans)
image(trans)

## 2. example: creating transactions from a matrix
a_matrix = matrix(
        c(1,1,1,0,0,
            1,1,0,0,0,
            1,1,0,1,0,
            0,0,1,0,1,
            1,1,0,1,1), ncol = 5)

## set dim names
dimnames(a_matrix) =  list(
            c("a","b","c","d","e"),
            paste("Tr",c(1:5), sep = ""))

a_matrix

## coerce
trans2 =  as(a_matrix, "transactions")
trans2

## example 3: creating transactions from data.frame
a_data.frame = data.frame(
            age = as.factor(c(6,8,7,6,9,5)),
            grade = as.factor(c(1,3,1,1,4,1)))
## note: all attributes have to be factors
a_data.frame

## coerce
```

```
trans3 = as(a_data.frame, "transactions")
image(trans3)

## 3. example creating from data.frame with NA
a_df = sample(c(LETTERS[1:5], NA),10,TRUE)
a_df = data.frame(X = a_df, Y = sample(a_df))
a_df
trans3 = as(a_df, "transactions")
trans3
as(trans3, "data.frame")
```

adult 자료

```
library(arules)
data(Adult)
str(Adult)
## Mine association rules: APRIOR알고리즘을 이용한 연관규칙의 탐색.
## 지지도 >= 0.5, 신뢰도 0.9 이상인 규칙들만 탐색함
rules = apriori(Adult,
        parameter = list(supp = 0.5, conf = 0.9, target = "rules"))
## 요약함수
summary(rules)

## 지지도 >=0.4 이상만
rules = apriori(Adult, parameter = list(support = 0.4))

## 좌측 아이템집합에 "sex"가 들어 있고 ##  규칙의 향상도가 0.3이상인
## 규칙만을 선택
rules.sub = subset(rules, subset = rhs %pin% "sex" & lift > 1.3)

## 선택된 규칙들을 보여줌
```

```
inspect(SORT(rules.sub)[1:3])
```

제 7 장

군집분석

제 1 절 군집분석에 대한 소개

모집단 또는 범주에 대한 사전 정보가 없는 경우 주어진 관측값들 사이의 거리 또는 유사성 을 이용하여 전체를 몇 개의 집단으로 그룹화하여 각 집단의 성격을 파악함으로써 데이터 전체의 구조에 대한 이해를 돕고자 하는 분석법

군집화

- 군집화의 기준
  동일한 군집에 속하는 개체는 여러 속성이 비슷하고, 서로 다른 군집에 속한 관찰치는 그렇지 않도록 구성
- 군집화를 위한 변수
  (예) 고객세분화
  - 인구통계적 변수 (성별, 나이, 거주지, 직업, 소득, 교육, 종교 등)
  - 구매패턴 변수 (상품, 주기, 거래액 등)
  - 생활패턴 변수 (라이프스타일, 성격, 취미, 가치관 등)

군집분석의 활용

고객 세분화

- 고객이 기업의 수익에 기여하는 정도를 통한 고객세분화
  - 우수고객의 인구통계적 요인, 생활패턴 파악
  - 개별고객에 대한 맞춤관리
- 고객의 구매패턴에 따른 고객세분화
  신상품 판촉, 교차판매를 위한 목표집단 구성

군집분석의 특징

- 그 기준의 설정, 즉 유사성이나 혹은 비유사성의 정의나 군집의 형태 등에 따라 다양한 방법이 존재
- 군집분석은 자료의 사전정보 없이 자료를 파악하는 방법으로, 분석자의 주관에 결과가 달라질 수 있음
- 특이값을 갖는 개체의 발견, 결측값의 보정 등에 사용될 수 있음
- 변수의 선택이 중요

제 2 절 거리

- 군집분석에서는 관측값들이 서로 얼마나 유사한지 또는 유사하지 않은지를 측정할 수 있는 측도가 필요
- 군집분석에서는 보통 유사성 (similarity) 보다는 비유사성 (dissimilarity) 을 기준으로 하며 거리 (distance) 를 사용
- 거리의 정의 : 두 점 $\boldsymbol{x}$ 와 $\boldsymbol{y}$ 의 거리 $d(\boldsymbol{x}, \boldsymbol{y})$ 는 다음을 만족한다.
  - $d(\boldsymbol{x}, \boldsymbol{y}) = 0 \Rightarrow \boldsymbol{x} = \boldsymbol{y}$
  - $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$
  - $d(\boldsymbol{x}, \boldsymbol{y})) = d(\boldsymbol{y}, \boldsymbol{x}))$
  - $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ (triangular inequality)

여러가지 거리

- 유클리드(Euclid) 거리

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{p} (x_i - y_i)^2 \right)^{1/2}$$

- Minkowski 거리

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{p} (x_i - y_i)^m \right)^{1/m}$$

- 표준화 거리

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{p} (x_i - y_i)^2 / s_i^2 \right)^{1/2}$$

- Mahalanobis 거리

$$d(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x}$$

범주형 자료의 거리

불일치 항목수

개체	성별	학력	출신지역
A	남자	고졸	경기
B	여자	고졸	전남
C	남자	대졸	경기

$d(A, B) = 2,\ d(A, C) = 1,\ d(B, C) = 3$

제 3 절 계층적 군집분석

- 가까운 관측값들 끼리 묶는 병합(agglomeration) 방법과 먼 관측값들을 나누어가는 분할(division) 방법

- 계층적 군집분석에서는 주로 병합 방법이 주로 사용

- 결과를 나무구조인 덴드로그램 (dendrogram) 을 통해 간단하게 나타낼 수 있고, 이를 이용하여 전체
- 군집들간의 구조적 관계를 쉽게 살펴볼 수 있음

병합방법

- 처음에 n 개의 자료를 각각 하나의 군집으로 취급
- n 개의 군집 중 가장 거리가 가까운 두 개의 군집을 병합하여 $n-1$ 개의 군집 형성
- $n-1$ 개의 군집 중 가장 가까운 두 군집을 병합하여 군집을 $n-2$ 개로 줄임
- 이를 반복하여 계속하여 군집의 수를 줄임
- 이 과정은 시작부분에는 군집의 크기는 작고 동질적이며, 끝부분에서는 군집의 크기는 커지고 이질적이 됨
- 군집들간의 거리를 측정하는 방법에 따라 다양한 종류의 방법이 있음
- 최단거리, 최장거리, 평균거리 방법 등

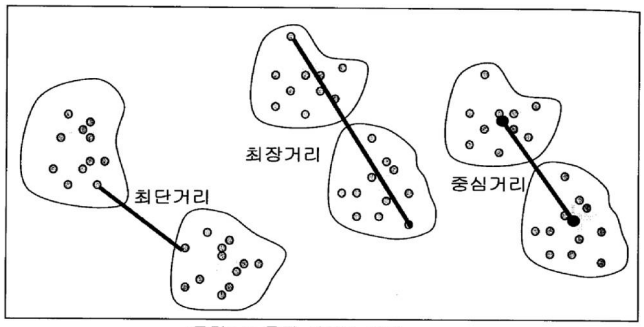


그림 7.1: 계층적 군집분석 예시

최단연결법 (Single Linkage Method)

- 두 군집 사이의 거리를 각 군집에서 하나씩 관측값을 뽑았을 때 나타날 수 있는 거리의 최소값으로 측정

- 같은 군집에 속하는 관측값은 다른 군집에 속하는 관측값에 비하여 거리가 가까운 변수를 적어도 하나는 갖고 있음
- 군집이 고리형태로 연결되어 있는 경우에는 부적절한 결과
- 고립된 군집을 찾는데 중점을 둔 방법

다음에 주어진 5개의 관측값에 대한 거리 행렬 (유사성 행렬)에 대하여 최단연결법으로 군집을 얻고 덴드로그램으로 나타내보자.

1	0				
2	7	0			
3	1	6	0		
4	9	3	8	0	
5	8	5	7	4	0
	1	2	3	4	5

1단계

- 거리 행렬에서 $d(1,3) = 1$ 이 최소이므로 관측값 1과 3을 묶어 군집 (1,3)을 만든다.
- 군집 (1,3)과 관측값 2,4,5와의 거리

$$d((2),(1,3)) = \min\{d(2,1), d(2,3)\} = d(2,3) = 6$$

$$d((4),(1,3)) = \min\{d(4,1), d(4,3)\} = d(4,3) = 8$$

$$d((5),(1,3)) = \min\{d(5,1), d(5,3)\} = d(5,3) = 7$$

를 구하여 다음과 같은 거리 행렬을 만든다.

(1,3)	0			
2	6	0		
4	8	3	0	
5	7	5	4	0
	(1,3)	2	4	5

2 단계

- 다음의 거리 행렬에서 d(2,4)=3 이 최소값을 가지므로 관측값 2 와 4 를 묶어 군집 (2,4) 를 만든다.
- 군집 (2,4) 와 군집 (1,3), (5) 와의 거리를 구한 후 거리 행렬을 다시 만든다.

$$\begin{aligned} d((2,4),(1,3)) &= \min\{d((2),(1,3)), d((4),(1,3))\} \\ &= d((2),(1,3)) = 6 \\ d((5),(2,4)) &= \min\{d(5,2), d(5,4)\} = d(5,4) = 4 \end{aligned}$$

(1,3)	0		
(2,4)	6	0	
5	7	4	0
	(1,3)	(2,4)	5

3 단계

- $d((5),(2,4)) = 4$ 이 최소값을 가지므로 군집 (2,4) 와 (5) 를 묶어 (2,4,5) 를 만든 후 $d((1,3),(2,4,5)) = d(2,3) = 6$ 을 이용하여 다음의 거리행렬을 얻는다.

(1,3)	0	
(2,4,5)	6	0
	(1,3)	(2,4,5)

4 단계 : 전체가 하나의 군집을 이룸

최장연결법 (Complete Linkage Method)

- 두 군집 사이의 거리를 각 군집에서 하나씩 관측값을 뽑았을 때 나타날 수 있는 거리의 최대값으로 측정
- 같은 군집에 속하는 관측치는 알려진 최대 거리보다 짧음
- 군집들의 내부 응집성에 중점을 둔 방법

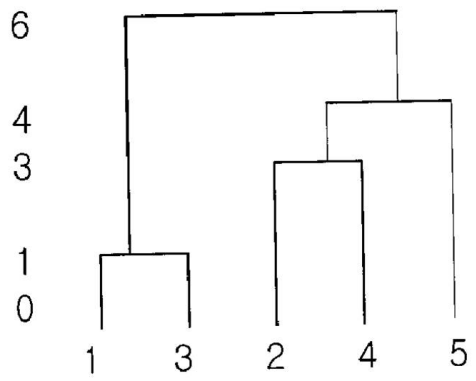


그림 7.2: 최단연결법 예제에 대한 덴드로그램

다음에 주어진 5개의 관측값에 대한 거리 행렬에 대하여 최장연결법으로 군집을 얻고 덴드로그램으로 나타내보자.

1	0				
2	7	0			
3	1	6	0		
4	9	3	8	0	
5	8	5	7	4	0
	1	2	3	4	5

1단계

- 1단계는 최단연결법의 1단계와 같다. 즉, 관측값 1과 3이 최단거리에 위치하고 이를 묶어서 새로운 군집 (1,3)을 만든다.
- 군집 (1,3)과 관측값 2,4,5와의 거리

$$d((2),(1,3)) = \max\{d(2,1), d(2,3)\} = d(2,1) = 7$$

$$d((4),(1,3)) = \max\{d(4,1), d(4,3)\} = d(4,1) = 9$$

$$d((5),(1,3)) = \max\{d(5,1), d(5,3)\} = d(5,1) = 8$$

를 구하여 다음과 같은 거리 행렬을 만든다.

(1,3)	0			
2	7	0		
4	9	3	0	
5	8	5	4	0
	(1,3)	2	4	5

2 단계

- 다음의 거리 행렬에서 d(2,4)=3 이 최소값을 가지므로 관측값 2 와 4 를 묶어 군집 (2,4) 를 만든다.
- 군집 (2,4) 와 군집 (1,3), (5) 와의 거리를 구한 후 거리 행렬을 다시 만든다.

$$\begin{aligned} d((2,4),(1,3)) &= \max\{d((2),(1,3)), d((4),(1,3))\} = d((4),(1,3)) = 9 \\ d((5),(2,4)) &= \max\{d(5,2), d(5,4)\} = d(2,4) = 5 \end{aligned}$$

(1,3)	0		
(2,4)	9	0	
5	7	5	0
	(1,3)	(2,4)	5

3 단계

- $d((5),(2,4)) = 4$ 이 최소값을 가지므로 군집 (2,4) 와 (5) 를 묶어 (2,4,5) 를 만든 후 $d((1,3),(2,4,5)) = d(1,4) = 9$ 을 이용하여 다음의 거리행렬을 얻는다.

(1,3)	0	
(2,4,5)	9	0
	(1,3)	(2,4,5)

4 단계 : 전체가 하나의 군집을 이룸

제 4 절 비계층적 군집분석

- 관측값들을 몇 개의 군집으로 나누기 위하여 주어진 판정기준을 최적화

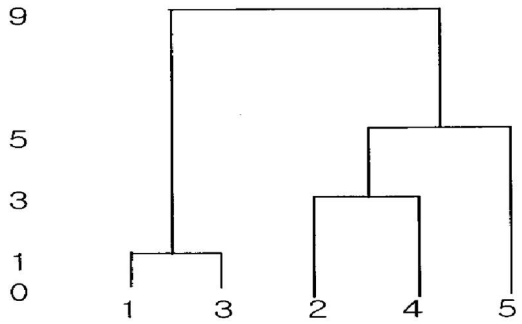


그림 7.3: 최장연결법 예제에 대한 덴드로그램

- 최적분리 군집분석이라고도 함
- 대표적인 비계층적 군집분석 방법이 K-평균 방법

K-평균 군집

- 사전에 결정된 군집수 K에 기초
- 전체 데이터를 상대적으로 유사한 K개의 군집으로 구분
- K-평균 군집법은 계층적 군집법에 비하여 계산량이 적음
- 대용량 데이터를 빠르게 처리할 수 있음

알고리즘

- 군집수 K를 결정
- 랜덤하게 초기 K개 군집의 중심을 선택
- 각 관측값을 그 중심과 가장 가까운 거리에 있는 군집에 할당하고 군집 중심을 새로 계산
- 위의 과정을 기존의 중심과 새로운 중심의 차이가 없을 때까지 반복

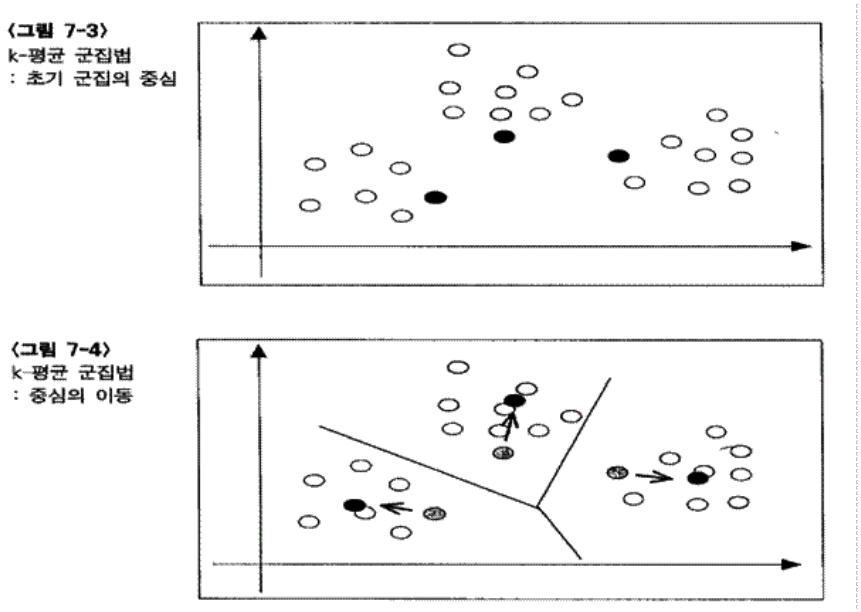


그림 7.4: K-평균 군집 알고리즘에 대한 예시

초기군집수의 결정

- K-평균 군집방법의 결과는 초기 군집수 K의 결정에 민감하게 반응
- 여러 가지의 K값을 선택하여 군집분석을 수행한 후 가장 좋다고 생각되는 K값을 이용
- 어떤 결과가 좋은가 하는 문제는 관측값간의 평균 거리(Within Sum of Squares)와 군집간의 평균거리(Between Sum of Squares)를 비교함으로써 수행
- 가장 좋은 방법은 자료의 시각화를 통한 최적 군집수의 결정인데, 자료의 시각화를 위하여는 차원의 축소가 필수적이고, 이를 위하여 주성분 분석방법(PCA)이 널리 사용됨
- 시각화가 어려운 경우에는 여러 가지 통계량을 사용하는데, 예를 들면, 각 그룹의 산포행렬의 행렬식을 최소로 하는 군집수를 찾음

K-평균 군집 방법의 변형

- K-평균방법의 단점

  - 군집이 겹치는 경우에 좋지 않음
  - 이상치에 민감
  - 각 관측값이 할당된 군집에 속하지 않을 불확실성에 대한 측정치 없음

- 대안으로 가우스 혼합모형 (Gaussian mixture model) 이 있음

가우스 혼합모형

- 주어진 군집수 k 에 대하여, 각 군집의 관측치의 분포가 미지의 평균과 분산을 따르는 정규분포 가정
- 자료를 가장 잘 분리할 수 있는 최적의 평균과 분산을 추정과 우도함수의 최대화의 두 단계를 반복적으로 구함 (Expectation-Maximization)
- 결과로 각 관측값에 대하여 그 관측값이 각 군집에 속할 확률을 계산
- 각 관측값을 가장 높은 확률을 갖는 집단으로 할당
- 이러한 방법을 소프트 군집화 (soft clustering) 이라 함

자기조직화지도

- 자기조직화지도 (Self-Organizing Map) 또는 Kohonen Map 이라 함
- 비슷한 개체들이 서로 이웃하는 위치에 오도록 배치
- K-평균 군집에 비해 계산량이 많음
- 그리드의 폭과 길이를 사전에 지정해야 함
- 비선형적 차원축소에 의한 다변량 개체들의 위상적 배열 (topological ordering)

제 5 절 군집분석의 고려사항

단위 / 가중치 문제

- 표준화

  - 군집분석은 자료 사이의 거리를 이용하여 수행되기 때문에 자료의 단위가 결과에 큰 영향을 미침 ⇒ 자료의 표준화
  - 표준화 : 각 변수의 관측값에서 평균을 빼고 표준편차로 나누는 것
  - 표준화된 자료는 모든 변수가 평균이 0 이고 표준편차가 1

- 가중치
  - 각 변수의 중요도가 다를 경우 가중치를 이용하여 각 변수의 중요도를 조절
  - 가중치는 대부분의 경우 단위변환(표준화)를 수행한 후 부여
  - 가중치에 대한 군집의 영향을 평가 하기 위하여는 여러 가지의 가중치에 대하여 군집분석의 결과를 비교

군집 평가

- 분석자의 주관에 의하여 결정되는 여러가지 사항들(예를 들면, 초기군집수, 가중치 등)이 군집분석의 결과에 어떻게 영향을 미치는가를 알아보기 위해서는 군집분석 결과의 평가가 필수적
- 사용되어진 거리의 측도를 이용하여 군집내의 거리의 평균과 군집간의 거리의 평균을 비교할 수 있음. 즉, 군집내의 거리의 평균이 군집간의 거리의 평균 보다 작을수록 좋음

제 6 절 군집분석의 응용 : 부정탐지

- 모터제조 공장에서 모터의 불량원인을 알고자 한다.
  - 정상적인 모터의 자료를 이용하여 군집분석을 수행하여 하나의 군집을 찾음
  - 새로운 모터와 이 군집과의 거리가 크면 이 새로운 모터를 불량모터라 의심
- 신용카드 사기, 보험료 과다 청구등 부정탐지(fraud detection)에 적용 가능

제 7 절 군집분석의 특징

- 탐색적인 기법: 주어진 자료에 대한 사전정보 없이 의미있는 자료구조를 찾아낼 수 있음
- 다양한 형태의 데이터에 적용가능: 거리만 잘 정의되면 모든 종류의 자료에 적용할 수 있음
- 분석방법의 적용이 쉬움
- 가중치와 거리 정의의 어려움
- 초기 군집수 k의 결정이 어려움
- 결과의 해석이 어려움

제 8 절 R 예제

R libraries

- `stat` hclust
- `cluster` Cluster Analysis Extended Rousseeuw et al.
- `clusterfly` Explore clustering interactively using R and GGobi
- `clustvarsel` Variable Selection for Model-Based Clustering
- `gclus` Clustering Graphics
- `mclust` Model-Based Clustering / Normal Mixture Modeling

거리 계산

```
x = matrix(rnorm(100), nrow=5)
dist(x)
dist(x, method= "manhatan")
```

표 7.1: R Packages

Function	Description	Package
hclust	Hierarchical Clustering	stats
kmeans	K-Means Clustering	stats
dist	distance matrix computed by using the specified distance measure	stats
cophenetic	Cophenetic Distances for a Hierarchical Clustering	stats
agnes	Agglomerative Nesting (Hierarchical Clustering)	cluster
clara	Clustering Large Applications	cluster
diana	**DI**visive **ANA**lysis Clustering	cluster
pam	Partitioning (clustering) of the data into k clusters "around medoids", a more robust version of K-means.	cluster
fanny	Computes a fuzzy clustering of the data into k clusters.	cluster
mona	MONothetic Analysis Clustering of Binary Variables	cluster
bannerplot	Plot Banner (of Hierarchical Clustering)	cluster
pltree	Clustering Trees - Generic Function	cluster
silhouette	Compute or Extract Silhouette Information from Clustering	cluster
Mclust	Model-Based Clustering	mclust
hc	Model-based Hierarchical Clustering	mclust
mclustBIC	BIC for Model-Based Clustering	mclust
plot.Mclust	Plot Model-Based Clustering Results	mclust
uncerPlot	Uncertainty Plot for Model-Based Clustering	mclust

```
dist(x, method= "maximum")


x = c(0, 0, 1, 1, 1, 1)
y = c(1, 0, 1, 1, 0, 1)
dist(rbind(x,y), method= "binary")
    x
y 0.4


hamming=function(x,y){sum(x != y)}


hamming(x,y)
[1] 2
```

계층적 군집분석

```
x = matrix(rnorm(100), nrow=5)
dist(x)
par(mfrow=c(2,2))
plot(h<-hclust(dist(x), method = "single"))
plot(h<-hclust(dist(x), method = "complete"))
plot(h<-hclust(dist(x), method = "average"))
plot(h<-hclust(dist(x), method = "centroid"),hang=-1)
```

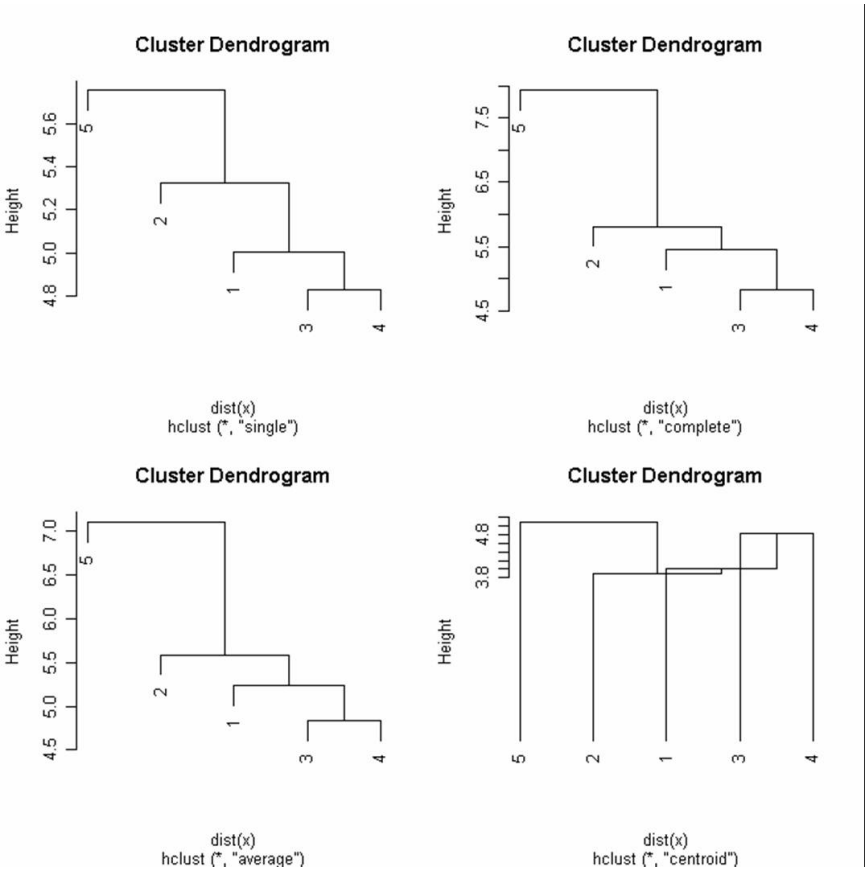


그림 7.5: 계층적 군집분석 예제의 결과

K-평균 군집분석

```
kmeans(x, centers, iter.max = 10, nstart = 1,
```

```
       algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"))

INPUT ARGUMENTS:
x :data matrix
centers : cluster의 개수 혹은 cluster 수에 해당되는 초기값 벡터들.
iter.max: 최대 반복수
nstart: If centers is a number, how many random sets should be chosen?
algorithm:

결과:
$cluster: A vector of integers indicating the cluster
$centers: A matrix of cluster centres.
$withinss: The within-cluster sum of squares for each cluster.
$size: The number of points in each cluster

    clusplot(x, rbinom(25,2, 0.5)+1, shade=F, color=T, lines=0)
```

K-medoids

```
Function: pam(x, k=#, diss=) :
    Partitioning Around Medoids (K-medoids Cluster Analysis)
    -the argument x can be either a matrix of data in which case
     diss=F, or x can be a dissimilarity matrix where diss=T.
     The parameter k, is the desired number of clusters
> data.pam = pam(data.diss, diss=T, k=3)
    # Clusters the data from the dissimilarity matrix
    #  data.diss into 3 groups around medoids.
```

제 9 절 SK telecom 기지국 grouping

배경

- 통신사의 서비스종류(음성, 문자, 동영상 등)가 다양함
- 기지국(수천개)별 서비스 이용패턴이 상이함.
- 서비스의 종류에 따라 다른 품질 요구수준(QoS)을 가짐
- 서비스와 관련된 기지국의 특성치를 바탕으로 유사한 기지국의 grouping이 요구됨

기지국 그룹핑을 위한 절차

- 기지국 특성을 (주로 용량) 결정하는 변수의 탐색: 총 12개의 변수 선택
  - 서비스별 가입자mix 평균 (HS/Browsing, Streaming, Download, USB 모뎀, Application)
  - 서비스별 가입자mix 표준편차 (HS/Browsing, Streaming, Download, USB 모뎀, Application)
  - 실최번시 Plysical Layer의 Forward Link Total Kbyte량 평균
  - 실최번시 Call Traffic 발신 시도호 평균
- 선정된 변수를 바탕으로 기지국 그룹핑: cluster analysis를 이용한 그룹핑
  - clustering algorithm: hierarchical + k-means
  - determine the number of clusters: 다음의 3가지 measure 이용
    - *Calinski and Harabasz (1974)* choose K to maximize

$$CH(K) = \frac{B(K)/(K-1)}{W(K)/(n-K)},$$

      where $B(K)$ is the sum of squares between cluster means.

    - *Hartingan (1975)* selects the smallest K if $H(K) \leq 10$ where

$$H(K) = \frac{W(K) - W(K+1)}{W(K+1)} \Big/ (n - K - 1).$$

* *Krzanowski and Lai (1985)*: choose K to maximize

$$KL(K) = \left| \frac{DIFF(K)}{DIFF(K+1)} \right|,$$

where $DIFF(K) = (K-1)^{2/p} W(K-1) - K^{2/p} W(K)$, and p is the dimension of variables.

- 그룹핑 결과에 대한 Reporting
  - 다차원 자료의 graphical display를 위해서 PCA 결과를 이용하여 first, second principal components를 두 축으로하는 평면에 도시
  - 그룹내의 기지국의 대표특성값을 계산(평균 및 표준편차)

PCA를 위한 이상치의 정도 결정

보통 다변량 이상치는 다변량 정규분포에 기반한 방법론을 이용한다. 가장 많이 사용되는 방법으로는 다변량자료를 표준화 시킨 후 다차원 자료의 center에서 각 자료점간의 Mahaanobis 거리를 잰다. Mahaanobis 거리는 다변량 정규분포인 경우 $\chi^2(p)$를 따르므로 적당한 신뢰도 하에서 χ^2-quantile을 계산하고 그 값 보다 크면 이상치로 간주한다. 아래는 9월 자료에 대하여 신뢰도 수준을 달리 할 경우에 대한 χ^2 quantile과 이상치로 판정된 자료의 비율이다. 실제 자료에서 계산된 Mahalanobis 거리의 quantile은 가장 오른쪽 column에 나와 있다.

	prob(%)	χ^2quantile	n	ratio	data quantile
1	95.00	21.02601	1664	0.4349190	170.65
2	97.50	23.33660	1571	0.4106116	397.02
3	99.00	26.21693	1460	0.3815996	2935.89
4	99.90	32.90949	1279	0.3342917	33899.78
5	99.99	39.13440	1139	0.2976999	

위의 표에서는 위 변수에 대한 Principal Component Analysis

Methods

- Clustering algorithm
  - k-Means
  - Hierarchical clustering
  - Model-based clustering
- Determination of the number of clusters.
- Output: numeric or graphical

제 8 장

기타 지도학습방법

제 1 절 k-근방 분류

시험자료점과 가까운 k 개의 훈련자료점의 y 값들을 비교하여 가장 많은 class 로 예측하는 방법

N_x 를 x 와 가까운 k 개의 Train 자료점들의 집합이라고 할 때 (이를 x 의 k-근방이라고 부름),

$$\hat{y}(x) = \arg \max_{l \in \{1,\ldots,L\}} \sum_{x_j \in N_x} I(y_j = l).$$

제 2 절 서포트 벡터 기계 (Support Vector Machines)*

$\{(x_i, y_i)\}_{i=1}^n$ 은 훈련표본으로서 $x_i \in R^l$ 이고 $y_i \in \{-1, +1\}$ 이다. 여기서 x_i 와 y_i 는 각각 입력변수와 출력변수라 불린다. $\langle \cdot, \cdot \rangle$ 을 R^l 상의 내적이라 하자. 분류는 훈련표본을 이용하여 $f(x) = \langle w, x \rangle + b$ 로 정의되는 선형 판별함수를 얻은 후, 그 부호를 이용하여 새로운 입력 x 에 대한 출력값을 예측하는 것으로 볼 수 있다. w 의 해를 얻기 위하여 서포트 벡터 기계는 다음과 같은 이차함수를 최적화한다.

$$\min_{w,b} \frac{1}{2}\langle w, w \rangle + C \sum_{i=1}^{n} \xi_i, \tag{8.1}$$

$$\text{subject to } y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots, n.$$

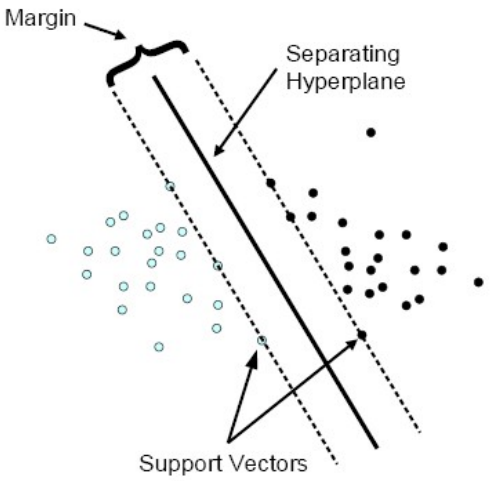


그림 8.1: 선형분리가능한 경우 서포트벡터기계

여기서 $C > 0$는 훈련오류와 벌점간의 상대적 크기를 조절하는 벌점항 모수이고 ξ_i는 slack 변수라 불린다. 흔히 (8.1)를 직접 최적화하는 대신 이에 대한 다음과 같은 쌍대 (dual) 목적함수를 최적화한다.

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{n} \alpha_i, \tag{8.2}$$

$$\text{subject to } \sum_{i=1}^{n} y_i \alpha_i = 0, 0 \le \alpha_i \le C, i = 1, \ldots, n.$$

(8.2)의 해를 $\hat{\alpha}_i, i = 1, \ldots, n$이라 하면 해가 되는 선형 판별함수는

$$\hat{f}(x) = \sum_{i=1}^{n} \hat{\alpha}_i y_i \langle x_i, x \rangle + \hat{b} \tag{8.3}$$

로 주어진다. 이 해가 0이 아닌 $\hat{\alpha}_i$ 값으로만 표현되므로 이러한 0이 아닌 $\hat{\alpha}_i$들을 대응되는 x_i들을 소위 서포트 벡터라 부른다. 여기서 $\hat{b}$는 Karush-Kuhn-Tucker 경계 조건들을 이용하여 결정된다. 비선형 분류 문제는 내적 $\langle \cdot, \cdot \rangle$ 대신 비선형 커널 $k(\cdot, \cdot)$을 이용하여 다룰 수 있다. 흔히 사용되는 비선형 커널로는 $k(x, y) = \exp(-\gamma \|x - y\|^2)$로 정의된 radial basis function(RBF) 커널이 있다. 여기서 $\gamma > 0$는 스케일 모수이다. 그러면 내적을 커널로 대체한 후 (8.2)의 쌍대 최적화 문제를 풀면 비선형 분류에서의 해는 (8.3)과 동일한 형태를 갖는다. 자세한 사항은 Vapnik (1998)을 참조하기 바란다.

제 3 절 앙상블기법

여러 분류모형에서 얻은 결과들에 다수결 원칙을 적용하여 최종 예측치를 결정하는 기법. 하나의 자료가 주어지면 붓스트랩(bootstrap) 표본추출 등에 의해 다수의 훈련자료를 생성하고 각 훈련자료에 동일한 알고리즘(예: 의사결정나무)으로 (또는 서로 다른 알고리즘으로) 모형을 생성한 후 그 결과를 결합하여 최종 예측을 함

배깅 (bagging)

알고리즘 (K 클래스 분류문제)

- Step 1: 표본 크기가 N 개인 자료 $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ 에서 B 개의 붓스트랩 표본 $\boldsymbol{X}^{*1}, \ldots, \boldsymbol{X}^{*B}$ 을 추출하고 각 붓스트랩 표본에 대하여 분류모형 $C(\boldsymbol{X}^{*1}, t), \ldots, C(\boldsymbol{X}^{*B}, t)$ 을 적합함
- Step 2: Step 1 에서 얻은 B 개의 모형들을 다음과 같이 결합

$$\hat{C}_{bag}(t) = \frac{1}{B} \sum_j C(\boldsymbol{X}^{*j}, t)$$

- Step 3: $\hat{C}_{bag}(t)$ 이 주는 K 개의 클래스 중 가장 큰 값을 갖는 클래스로 예측

참고: Breiman 은 반복회수 B 로 50 정도가 적당하다고 제안
배깅의 특징

- 의사결정나무, 신경망, 로지스틱 회귀 등 다양한 방법에 적용 가능
- 의사결정나무에 적용했을 때 가장 효과적
- 예측오류의 상승폭이 큰 불안정한 방법(예: 의사결정나무)에 적용할 경우 예측 정확도의 향상
- 의사결정나무에 배깅을 적용할 때는 가지치기를 생략해도 됨
- 배깅을 이용한 경우 단일 의사결정나무처럼 해석할 수 없는 단점

부스팅 (boosting)

- 부스팅과 배깅의 차이: 붓스트랩 표본추출 확률
  - 배깅: 원 훈련자료의 각 관측치가 표본으로 뽑힐 확률이 항상 일정
  - 부스팅: 원 훈련자료의 각 관측치에 대한 표본 추출 확률이 단계별로 달라짐
- 각 모형에서 예측결과 결합시 가중평균을 사용
  - 배깅: 가중치가 동일
  - 부스팅: 모형의 정확도가 높을수록 가중치가 커짐

범핑 (bumping)

붓스트랩 표본을 추출하여 모형을 적합하는 방법으로 배깅에서는 각 모형을 결합하여 예측하지만 범핑의 경우는 여러 모형중 가장 예측력이 좋은 모형 하나만을 선택하고 이 모형을 이용하여 최종 예측을 함

제 4 절 신용평점표

- 고객의 신용 상태(우량/불량)을 예측하기 위해 만들어진 점수 할당표
- 각 특성들은 적절한 속성으로 나뉘고 각 속성에는 점수가 부여됨
- 각 고객의 신용평점은 각 속성별로 점수들을 합하여 산출됨
- 평점표 개발은 과거자료에서 변수를 추출하고 변환하는 단계와 모델링 과정을 거치는데 주로 로지스틱 회귀모형을 이용
- 각 설명변수는 가변수로 변환됨. 연속형 변수는 이산형으로 이산형 변수는 재 그룹핑을 통한 변환과정을 거침

신용평점의 종류

- 일반적 점수(generic score)

  - 신용회사의 정보만을 이용하여 고객의 미래 지불 능력을 예측하기 위해 만든 평점표
  - 신용정보회사가 보유한 과거 지불행위 정보, 신용거래 정보, 생활 신용정보 등이 이용되고 인구통계자료의 활용은 불가능

- 고객 점수(custom score)
  - 신청서의 모든 자료를 이용하여 금융기관이 직접 개발한 평점표
  - 신용회사 정보뿐 아니라 인구통계자료를 활용하며 고객에 대한 충분한 이해가 가능한 장점

신용평점표 작성

- 변수 선택
  - 지니계수 등을 이용하여 유의한 변수를 선택
  - 단변량적인 비교로 인한 변수 상호간의 영향력을 고려하지 못함
  - 전진선택법 등의 변수 선택법을 사용하기도 함

- 연속형 설명변수의 범주화
  - 구간의 개수를 정하고 각 구간에 속한 개체의 수가 비슷하도록 함
  - 범주화를 평가하는 방법으로 카이제곱 통계량을 사용하며 값이 클수록 범주화가 효과적
  - 범주화된 변수들을 가변수화

- 로지스틱 회귀모형 적용
  각 변수의 계수값은 고객이 얻게 되는 점수로 이 값을 이용하여 평점표 작성

- 평가
  ROC 곡선, Mahalanobis 거리, Kolmogorov-Smirnov 거리 등

- 범주의 조정
  각 가변수의 계수값을 조정

제 5 절 RFM 모형

- 최근성(recency), 구매빈도(frequency), 구매금액(monetary) 등 고객의 수익기여도를 나타내는 세 가지 지표들의 선형결합으로 구한 점수

$$RFM = A \cdot Recency + B \cdot Frequency + C \cdot Monetary$$

- 다중회귀분석을 통해 계수 A, B, C 추정
- 효율적인 고객관리를 위해 고객을 세분화하고 채산성이 있는 고객을 대상으로 마케팅함으로써 매출액 증대보다는 수익을 창출하기 위한 모형
- 최근성, 구매빈도, 구매금액에 대한 예
  - 최근성: 지난 3, 6, 12, 18, 24 개월 내에 거래가 있음: 5, 4, 3, 2, 1 점
  - 구매빈도: 지난 24 개월 내에 거래가 있었고 각 거래 마다 0.5 점 (총점은 5 점 이내로 제한)
  - 구매금액: 화폐단위로 1 점씩 계산하되 상한을 둠 (이상점의 영향)

RFM 점수 산출 예

메일 발송할 목표 고객의 선정 예

- Recency 를 기준으로 고객을 정렬한 후 상위 20% 고객에 5 점을 부여하고 다음 상위 20% 의 고객에 4 점을 부여. 고객의 Recency 점수가 5, 4, 3, 2, 1 점이 되도록
- Frequency 와 Monetary 도 Recency 와 동일한 방식으로 점수 부여
- 전체 125 개의 집단으로 세분화되고 RFM 점수는 555, 554, ..., 111
- RFM 점수에 따른 메일 발송 수와 응답자 수, 응답율에 대한 표 작성
- 고객 응답율 예측

RFM 모형의 통계적 해석

- 다중회귀분석

  가장 많이 쓰임. 독립변수: R, F, M. 종속변수: 고객응답율

$$Y_i = \alpha + w_1 R_{ij} w_2 F_{ij} + w_3 M_{ij}$$

- 신경망
- 확률적 RFM 모형
  - 고객의 구매행위를 구매확률과 구매액으로 모형화
  - 마케팅 전략 수립이 용이

제 6 절 R 예제

k-근방 분류

```
library(class)
data(iris)
y<-iris[,5]
tr.idx<-sample(length(y), 75)
x.tr <- iris[tr.idx,-5]
x.te <- iris[-tr.idx,-5]
m1<-knn(x.tr, x.te, y[tr.idx], k = 3)
# k=number of neighbours considered.
```

서포트 벡터 기계*

```
library(e1071)
data(iris)
attach(iris)
```

```
## classification mode
# default with factor response:
m2 <- svm(Species~., data = iris, kernel="linear")
plot(m2, iris, Petal.Width ~ Petal.Length,
     slice = list(Sepal.Width = 3, Sepal.Length = 4))

print(model)
summary(model)
    Call:
    svm.default(x = x, y = y)

    Parameters:
    SVM-Type:  C-classification
    SVM-Kernel:  radial
        cost:  1
        gamma:  0.25

    Number of Support Vectors:  51
    ( 8 22 21 )

    Number of Classes:  3
    Levels:
    setosa versicolor virginica

# test with train data
pred <- predict(model, x)
# (same as:)
pred <- fitted(model)

# Check accuracy:
```

```
table(pred, y)
```

```
               y
    pred         setosa versicolor virginica
    setosa         50          0         0
    versicolor      0         48         2
    virginica       0          2        48
```

```
# compute decision values and probabilities:
pred <- predict(model, x, decision.values = TRUE)
pred, "decision.values")[1:4,]
```

```
  setosa/versicolor setosa/virginica versicolor/virginica
1          1.196152         1.091757            0.6705626
2          1.064621         1.056185            0.8479934
3          1.180842         1.074542            0.6436474
4          1.110699         1.053012            0.6778595
```

```
# visualize (classes by color, SV by crosses):
plot(cmdscale(dist(iris[,-5])),
     col = as.integer(iris[,5]),
     pch = c("o","+")[1:150 %in% model$index + 1])
```

배깅

```
#single tree and bagging tree
library(tree)
data(iris)
y<-iris[,5]
tr.idx<-sample(length(y), 75)
x.tr <- iris[tr.idx,]
x.te <- iris[-tr.idx,-5]
```

```
++++++++++++++++++++++++++++++++++++++++++++++
m0<-tree(Species~., data=x.tr)
m0.cv<-cv.tree(m0, FUN=prune.misclass)
    $size
    [1] 4 3 2 1
    $dev
    [1]  5  4 21 45
    $k
    [1] -Inf    1   18   24

m0.p<-prune.tree(m0, best=3)
mean(predict(m0.p, x.te, type="class") != y[-tr.idx])
    [1] 0.09333333
++++++++++++++++++++++++++++++++++++++++++++++

nbag<-20
m<-list()
n<-length(y)
for(i in 1:nbag)
{
    s.idx<-sample(1:n, n, replace=T)
    m[[i]]<-tree(Species~., data=x.tr[s.idx,])
}
pred<-sapply(m, FUN=function(m1)
                predict(m1, newdata=x.te, type="class")
            )

pred.bag<-apply(pred, 1, FUN=function(x)
                names(table(x))[which.max(table(x))]
```

```
            )
cat("Bagged Error =", mean(y[-tr.idx]!=pred.bag), "\n")
    Bagged Error = 0.02666667
```

부스팅

```
library(boost)
data(leukemia, package = "boost")

## Dividing the leukemia dataset into training and test data
xlearn <- leukemia.x[c(1:20, 34:38),]
ylearn <- leukemia.y[c(1:20, 34:38)]
xtest  <- leukemia.x[21:33,]
ytest  <- leukemia.y[21:33]

## Classification with adaboost
fit <- adaboost(xlearn, ylearn, xtest, presel=50, mfinal=200)
plot.boost<-function (boost.out, resp, mout = ncol(boost.out), ...)
{
  mcra <- apply(((boost.out > 0.5) * 1) != resp, 2, mean)
  mini <- which.min(mcra)
  mcrs <- round(min(mcra), 4)
  mcrf <- round(mean(((boost.out[, mout] > 0.5) * 1) != resp),
      4)
  cat("\n")
  cat("Minimal mcr:",mcrs,"achieved after",mini,"boosting step(s)\n")
  cat("Fixed mcr:  ",mcrf,"achieved after",mout,"boosting step(s)\n")
  xax <- "Boosting steps"
  yax <- "Error rate"
  plot(mcra, xlab = xax, ylab = yax, type = "l", ...)
}
```

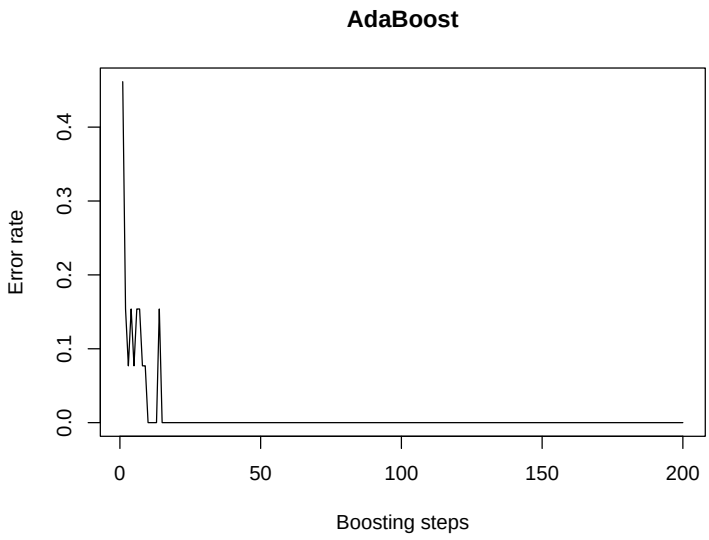


그림 8.2: AdaBoost 결과

```
pdf("adaboost.pdf")
plot.boost(fit, ytest, main="AdaBoost")
dev.off()
```